

TPC Benchmark™ C
Full Disclosure Report
for
IBM @server xSeries 360
using
Microsoft SQL Server 2000 Enterprise Edition
and
Windows 2000 Advanced Server

TPC-C Version 5.0

Submitted for Review
July 17, 2002



First Edition - July 2002

THE INFORMATION CONTAINED IN THIS DOCUMENT IS DISTRIBUTED ON AN AS IS BASIS WITHOUT ANY WARRANTY EITHER EXPRESSED OR IMPLIED. The use of this information or the implementation of any of these techniques is the customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

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

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

© Copyright International Business Machines Corporation 2002. All rights reserved.

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

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

Trademarks

IBM, xSeries and the e-business logo are trademarks or registered trademarks of International Business Machines Corporation.

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

Notes

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

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

Abstract

IBM Corporation conducted the TPC Benchmark™ C on the IBM @server xSeries 360 configured as a client/server system. This report documents the full disclosure information required by the TPC Benchmark™ C Standard Specification, Revision 5.0, including the methodology used to achieve the reported results. All testing fully complied with this revision level.

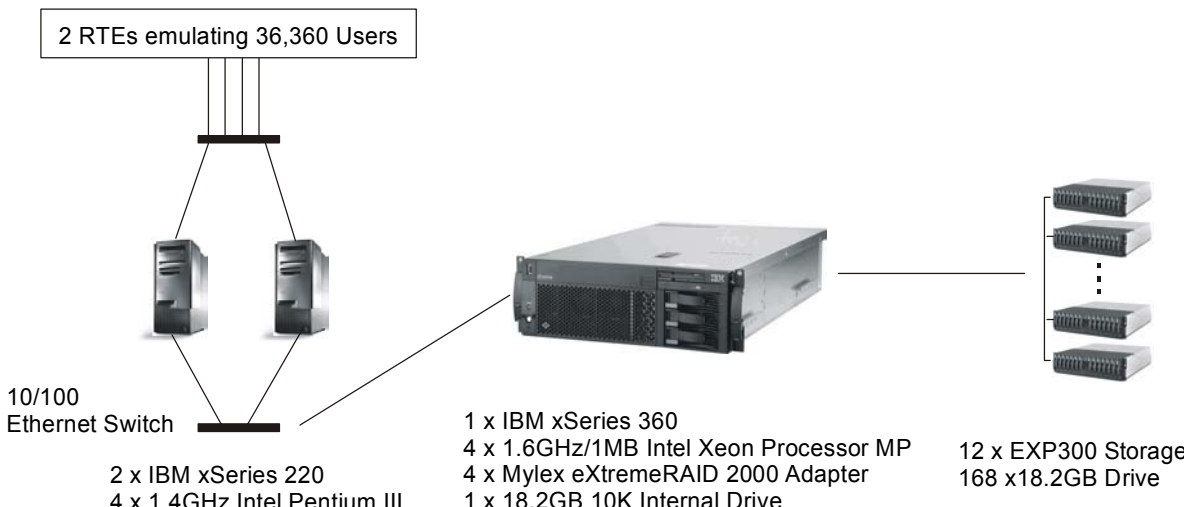
The software used on the xSeries 360 system includes Microsoft** Windows** 2000 Advanced Server operating system and Microsoft SQL Server 2000 Enterprise Edition database.

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

The benchmark results are summarized in the following table.

Hardware	Software	Total System Cost	tpmC	\$/tpmC	Total Solution Availability Date
IBM @server xSeries 360	Microsoft SQL Server 2000 Enterprise Edition Microsoft Windows 2000 Server	\$204,271	45,230.03	\$4.52	August 30, 2002

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

		IBM @server xSeries 360 c/s with Microsoft SQL Server 2000		TPC-C Rev. 5.0	
				Report Date: July 17, 2002	
Total System Cost		TPC-C Throughput		Price/Performance	Availability Date
\$204,271		45,230.03 tpmC		\$4.52 / tpmC	August 30, 2002
Processors	Database Manager	Operating System	Other Software		Number of Users
Database Server: 4 Xeon Processor MP 1.6GHz Client: 4 Pentium III 1.4GHz	Microsoft® SQL Server 2000 Enterprise Edition	Microsoft Windows® 2000 Advanced Server	Microsoft Visual C++ 6.0 Win32 Microsoft COM+		36,360
2 RTEs emulating 36,360 Users  10/100 Ethernet Switch 2 x IBM xSeries 220 4 x 1.4GHz Intel Pentium III 1 x IBM xSeries 360 4 x 1.6GHz/1MB Intel Xeon Processor MP 4 x Mylex eXtremeRAID 2000 Adapter 1 x 18.2GB 10K Internal Drive 12 x EXP300 Storage 168 x 18.2GB Drive					
System Component	Qty	Server:	Qty	Each Client:	
Processors	4	1.6GHz Xeon Processor MP w/1MB L3 Cache	2	1.4GHz Pentium III w/512KB L2 Cache	
Cache					
Memory	8	1GB ECC DDR SDRAM RDIMM	1	128MB	
			2	256MB	
Disk Controllers	4	Mylex eXtremeRAID 2000 Adapter with 32MB	1	Ultra160 SCSI Interface	
Disk Drives	169	18.2GB (10000 rpm)	1	18.2GB Hard Disk	
Total Storage		2861.17GB			
Tape Drive	1	20/40GB SCSI Tape Drive			

Numerical Quantities Summary			
MQTh, Computed Maximum Qualified Throughput:			45,230.03 tpmC
Response Times (in seconds)	Average	Maximum	90 %-tile
New-Order	0.43	5.28	0.60
Payment	0.27	3.62	0.43
Delivery	0.19	0.71	0.34
Stock Level	2.04	4.27	3.15
Order Status	0.29	5.23	0.44
Delivery (Deferred)	0.48	1.44	0.72
Menu	0.20	0.81	0.35
Transaction Mix (in percent of total transactions)		Total Occurrences	Percent
New-Order		5,427,604	44.87
Payment		5,206,759	43.04
Delivery		488,584	4.04
Stock-Level		486,682	4.02
Order Status		486,719	4.02
Emulation Delay (in seconds)		Response Time	Menu
New-Order		0.1	0.1
Payment		0.1	0.1
Delivery		0.1	0.1
Stock-Level		0.1	0.1
Order Status		0.1	0.1
Keying/Think Times (in seconds)	Average	Minimum	Maximum
New Order	18.01 / 12.05	18.00 / 0.00	18.10 / 120.51
Payment	3.01 / 12.06	3.00 / 0.00	3.10 / 120.51
Delivery	2.01 / 5.06	2.00 / 0.00	2.10 / 50.50
Stock Level	2.01 / 5.06	2.00 / 0.00	2.10 / 50.50
Order Status	2.01 / 10.05	2.00 / 0.00	2.11 / 100.51
Test Duration			
Ramp-up time			33 minutes
Measurement interval			120 minutes
Number of transactions (all types) completed in measurement interval			12,584,921
Ramp-down time			6 minutes
Number of checkpoints in measurement interval			4
Checkpoint interval			30 minutes

Table of Contents

Abstract	3
Numerical Quantities Summary	6
Preface	11
General Items	12
Application Code Disclosure and Definition Statements	12
Benchmark Sponsor	12
Parameter Settings	12
Configuration Diagrams	12
Clause 1: Logical Database Design Related Items	14
Table Definitions	14
Physical Organization of the Database	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: Transaction and System Properties Related	
Items	17
Atomicity Requirements	17
Consistency Requirements	17
Isolation Requirements	18
Durability Requirements	18
Clause 4: Scaling and Database Population Related	
Items	20
Cardinality of Tables	20
Distribution of Tables and Logs	21
Database Model Implemented	21
Partitions/Replications Mapping	21
60-Day Space Requirement	21
Clause 5: Performance Metrics and Response Time	
Related Items	22
Measured tpmC	22
Response Times	22
Keying/Think Times	22
Response Time Frequency Distribution Curves	23
Performance Curve for Response Time vs. Throughput	25
New Order Think Time Distribution	26
Throughput vs. Elapsed Time	26
Steady State Methodology	27
Work Performed during Steady State	27
Checkpoints	27
Measurement Interval	27
Transaction Mix	27
Percentage of Total Mix	28
Number of Checkpoints	28

Clause 6: SUT, Driver and Communication Definition

Related Items	29
Description of RTE	29
Emulated Components	29
Benchmarked and Targeted System Configuration	
Diagrams	29
Network Configuration	29
Network Bandwidth	29
Operator Intervention	29
Clause 7: Pricing Related Items	30
Hardware and Software Components	30
Availability Date	30
Measured tpmC	30
Country-Specific Pricing	30
Usage Pricing	30
System Pricing	31
Clause 9: Audit Related Items	32
Auditor	32
Availability of the Full Disclosure Report	32
<i>Attestation letter</i>	33
Appendix A: Source Code	35
<i>client_utils.c</i>	35
<i>client_utils.h</i>	36
<i>dlldata.c</i>	37
<i>error.h</i>	37
<i>install.C</i>	39
<i>install.h</i>	47
<i>install.rC</i>	47
<i>install_com.cpp</i>	50
<i>license.txt</i>	52
<i>mon_client.c</i>	54
<i>mon_client.h</i>	57
<i>readme.txt</i>	57
<i>Readregistry.cpp</i>	57
<i>Readregistry.h</i>	58
<i>Resource.h</i>	58
<i>Resource_tpcc_rc.H</i>	59
<i>rtetime.h</i>	59
<i>spinlock.h</i>	59
<i>tpcc.cpp</i>	60
<i>tpcc.def</i>	83
<i>tpcc.h</i>	83
<i>tpcc.rc</i>	85
<i>tpcc_com.cpp</i>	86
<i>tpcc_com.h</i>	87
<i>tpcc_com_all.dsp</i>	89
<i>tpcc_com_ps.def</i>	90
<i>tpcc_com_ps.h</i>	90
<i>tpcc_com_ps.idl</i>	92
<i>tpcc_com_ps_i.c</i>	93
<i>tpcc_com_ps_p.c</i>	94
<i>tpcc_dblib.cpp</i>	114
<i>tpcc_dblib.h</i>	124
<i>tpcc_enc.cpp</i>	125
<i>tpcc_enc.h</i>	127

<i>tpcc_odbc.cpp</i>	128
<i>tpcc_odbc.h</i>	135
<i>tpcc_tux.cpp</i>	137
<i>tpcc_tux.h</i>	139
<i>trans.h</i>	140
<i>tuxapp.cpp</i>	141
<i>tuxapp.h</i>	145
<i>tuxmain.c</i>	145
<i>txn_base.h</i>	146
<i>txnlog.h</i>	146
<i>webclnt.dsp</i>	149
<i>webclnt.dsw</i>	150
Stored Procedures	151
<i>neword.sql</i>	151
<i>payment.sql</i>	154
<i>ordstat.sql</i>	155
<i>delivery.sql</i>	156
<i>stocklev.sql</i>	157
<i>version.sql</i>	158
<i>null-txn.sql</i>	158
Appendix B: Database Design	162
Database Build	162
<i>backup.sql</i>	162
<i>backupdev.sql</i>	162
<i>createdb.sql</i>	162
<i>dbopt1.sql</i>	163
<i>dbopt2.sql</i>	163
<i>idxcuscl.sql</i>	163
<i>IDXCUSnC.SQL</i>	164
<i>idxdiscl.sql</i>	164
<i>idxitmcl.sql</i>	164
<i>idxnodcl.sql</i>	164
<i>idxodlcl.sql</i>	164
<i>idxordcl.sql</i>	165
<i>idxordnc.sql</i>	165
<i>idxstkcl.sql</i>	165
<i>idxwarcl.sql</i>	165
<i>removedb.sql</i>	166
<i>restore.sql</i>	166
<i>RunSQLCfg.sql</i>	166
<i>sqlshutdown.sql</i>	166
<i>tables.sql</i>	166
<i>Verify_TpccLoad.sql</i>	168
<i>version.sql</i>	169
Load Source Code	169
<i>getargs.c</i>	169
<i>random.c</i>	171
<i>strings.c</i>	172
<i>time.c</i>	175
<i>tpcc.h</i>	175
<i>tpccldr.c</i>	176
<i>tpccldr.mak</i>	200
Appendix C: Tunable Parameters	204
Microsoft Windows 2000 Server Configuration Parameters	204

Microsoft Windows 2000 Server Services	204
<i>Microsoft SQL Server Startup Parameters</i>	204
<i>Microsoft SQL Server 2000 Configuration Parameters</i>	204
<i>Microsoft Windows 2000 Server System Information Report</i>	205
Disk Controller Configuration Parameters	225
<i>Mylex eXtremeRAID 2000 Adapter 1</i>	225
<i>Mylex eXtremeRAID 2000 Adapter 2</i>	228
<i>Mylex eXtremeRAID 2000 Adapter 3</i>	231
<i>Mylex eXtremeRAID 2000 Adapter 4</i>	235
Client Configuration Parameters	238
<i>Microsoft Windows 2000 Server System Information Report</i>	238
<i>COM+ Settings</i>	259
RTE Input Parameters	260
Appendix D: 60-Day Space	268
Appendix E: Third-Party Quotations	269

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 5.0.

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

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

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

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

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

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

General Items

Benchmark Sponsor

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

This benchmark was sponsored by International Business Machines Corporation.

Application Code Disclosure and Definition Statements

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

Appendix A contains all source code implemented in this benchmark.

Parameter Settings

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

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

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

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

Configuration Diagrams

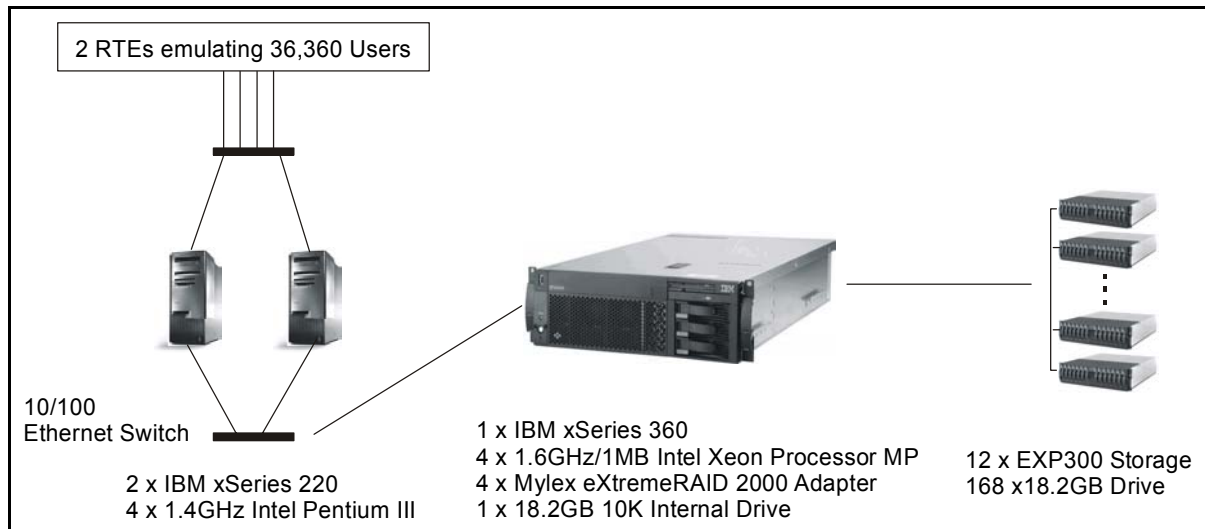
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences.

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

The Remote Terminal Emulator (RTE) used for these TPC Benchmark C tests is the Microsoft BenchCraft RTE. Under Version 5.0, the components of the configuration being emulated by the RTE are the workstations and the Ethernet hubs. Appendix C contains a listing of the RTE scripts and inputs used in the benchmark testing.

The benchmarked configuration used two IBM xSeries 220 system as the clients, which executed the terminal I/O and submitted transactions to COM+ servers, which are also running on the clients. These COM+ servers forwarded the transaction requests to the server, and returned the results to the RTE. Microsoft SQL Server 2000 Enterprise Edition is the DBMS executing on the server

Measured and Priced Configuration



Clause 1: Logical Database Design Related Items

Table Definitions

Listings must be provided for all table definition statements and all other statements used to set up the database.
Appendix B contains the code used to define and load the database tables.

Physical Organization of the Database

The physical organization of tables and indexes within the database must be disclosed.
Physical space was allocated to Microsoft SQL Server on the server disks as detailed in Figure 4-2.

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

All insert and delete functions were fully operational during the running of the benchmark. The space required for an additional 5 percent of the initial table cardinality was allocated to Microsoft SQL Server 2000 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 seeds and offsets for the random number generator were collected and verified to be different for each driver. The auditor selected samples of the generated numbers from the database. The samples were verified to have no discernible patterns.

Screen Layout

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

All screen layouts followed the TPC Benchmark C Standard Specification.

Terminal Verification

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

The auditor verified terminal features by direct experimentation. The benchmarked configuration uses Microsoft Internet Explorer 5.0 and HTML scripts as the terminal interface.

Intelligent Terminals

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

The terminals emulated in the priced configuration are IBM PC desktop computer systems. All processing of the input/output screens was handled by the xSeries 220 clients. The screen input/output was managed via HTML strings that comply with the HTML Version 2.0 specification. A listing of the code used to implement the intelligent terminals is provided in Appendix A. All data manipulation was handled by the xSeries 360 database server.

Transaction Profiles

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

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

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

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

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

Table 2-1. Transaction Statistics

New Order	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	85.01
Remote warehouse payment transactions	14.99
Non-Primary Key Access	
Payment transactions using C_LAST	59.98
Order-Status transactions using C_LAST	60.06
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.87
Payment	43.04
Delivery	4.04
Stock Level	4.02
Order Status	4.02

Deferred Delivery Mechanism

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

The deferred delivery operation is queued by making an entry in an array within the application process (tpcc.dll) running on the client. Background threads within the application asynchronously process the queued delivery transactions.

The source code is listed in Appendix A.

Clause 3: Transaction and System Properties Related Items

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

Atomicity Requirements

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

All ACID tests were conducted according to specification.

Completed Transactions

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

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

Aborted Transactions

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

1. The Payment application code was changed to execute a rollback of the transaction instead of performing the commit.
2. Using the balance, BALANCE_2, from the CUSTOMER table retrieved for the completed transaction, the Payment transaction was executed for the Customer, District and Warehouse used in step 1 of section 3.1.1. The transaction rolled back due to the change in the application code from step 1.
3. The balance was retrieved again for the Customer used for step 2, giving BALANCE_3. It was verified that BALANCE_2 was equal to BALANCE_3.

Consistency Requirements

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

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

Isolation Requirements

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

Isolation tests one through seven were run 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 and placed in files. The auditor reviewed the results and verified that the isolation requirements had been met.

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

Case A was followed for Isolation test seven.

Durability Requirements

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

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

Loss of Data Test

The following steps were successfully performed to pass the Durability test of failure of a disk unit with database tables:

1. The contents of the database were backed up to several database dump devices during the initial database load. There were no dump devices on the disk array from which a drive was removed as part of this test.
2. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID for all rows in the district table giving SUM1.
3. A test was started with 3,400 users (10 percent of the database) submitting transactions.
4. A disk containing a portion of each of the tables in the tpcc database was removed causing SQL Server to report errors accessing that device.
5. The run was aborted and SQL Server was restarted. Upon restart, the database tpcc reported numerous errors relating to the failed database device.
6. The transaction log was dumped to disk and the failed disk was replaced with a spare disk and was recovered.
7. The database was recovered and restored from the backup dump devices. Afterwards, the transaction log was applied to the database.
8. Step 2 was repeated to obtain the current count of the total number of orders giving SUM2.
9. 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.
10. Consistency Condition 3 was verified.

Combined Loss of Log and Loss of System Test (Instantaneous Interruption and Loss of Memory)

1. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID for all rows in the district table giving SUM1.
2. A test was started under full load with all users submitting transactions.
3. One of the log disk drives was removed. Since the log disk was mirrored, SQL Server continued to process transactions without interruption.
4. The test continued under full load with all users submitting transactions. A checkpoint was issued, and the system continued to run for another 5 minutes.
5. The server under test was powered off, which removed power from the system and the memory.
6. The server was powered on again.
7. SQL Server was started to initiate automatic recovery from its log.
8. Step 1 was repeated to obtain the current count of the total number of orders giving SUM2.
9. 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.

Clause 4: Scaling and Database Population Related Items

Cardinality of Tables

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

The database was originally built with 3,636 warehouses, and the audited run used all 3,636 warehouses.

Table 4-1. Initial Cardinality of Tables

Table Name	Rows
Warehouse	3,636
District	36,360
Item	100,000
New Order	32,724,000
History	109,080,000
Orders	109,080,000
Customer	109,080,000
Order Line	1,090,800,831
Stock	363,600,000
Inactive Warehouses	0

Distribution of Tables and Logs

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

Figure 4-2 depicts the database configuration of the tested system to meet the 8-hour steady state requirement.

Figure 4-2. Data Distribution for the Benchmarked Configuration

Controller	Drives	Partition	Size	Use
1	42 - 18.2GB	E: Y:	80000MB 150000MB (NTFS)	Customer and Stock Backup 1
2	42 - 18.2GB	F: Z:	80000MB 150000MB (NTFS)	Customer and Stock Backup 2
3	42 - 18.2GB	G:	80000MB	Customer and Stock
4	30 - 18.2GB	H:	100000MB	Misc.
4	12 - 18.2GB	I:	100000MB	Logfile

Database Model Implemented

A statement must be provided that describes:

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

Microsoft SQL Server 2000 Enterprise Edition is a relational database. The interface used was Microsoft SQL Server stored procedures accessed with Remote Procedure Calls embedded in C code using the Microsoft DBLIB interface.

Partitions/Replications Mapping

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

The database was neither partitioned nor replicated.

60-Day Space Requirement

Details of the 60-day space computations, along with proof that the database is configured to sustain 8 hours of growth for the dynamic tables (Order, Order-Line, and History) must be disclosed (see Clause 4.2.3).

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

Clause 5: Performance Metrics and Response Time Related Items

Measured tpmC

Measured tpmC must be reported.

Measured tpmC: 45,230.03 tpmC

Price per tpmC: \$4.52 per tpmC

Response Times

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

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

Table 5-1. Response Times in Seconds

Transaction Type	Average	Maximum	90 %-tile
New-Order	0.43	5.28	0.60
Payment	0.27	3.62	0.43
Delivery	0.19	0.71	0.34
Stock Level	2.04	4.27	3.15
Order Status	0.29	5.23	0.44
Delivery (Deferred)	0.48	1.44	0.72
Menu	0.20	0.81	0.35

Keying/Think Times

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

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

Table 5-2. Keying/Think Times

Transaction Type	Average	Minimum	Maximum
New-Order	18.01 / 12.05	18.00 / 0.00	18.10 / 120.51
Payment	3.01 / 12.06	3.00 / 0.00	3.10 / 120.51
Delivery	2.01 / 5.06	2.00 / 0.00	2.10 / 50.50
Stock Level	2.01 / 5.06	2.00 / 0.00	2.10 / 50.50
Order Status	2.01 / 10.05	2.00 / 0.00	2.11 / 100.51

Response Time Frequency Distribution Curves

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

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

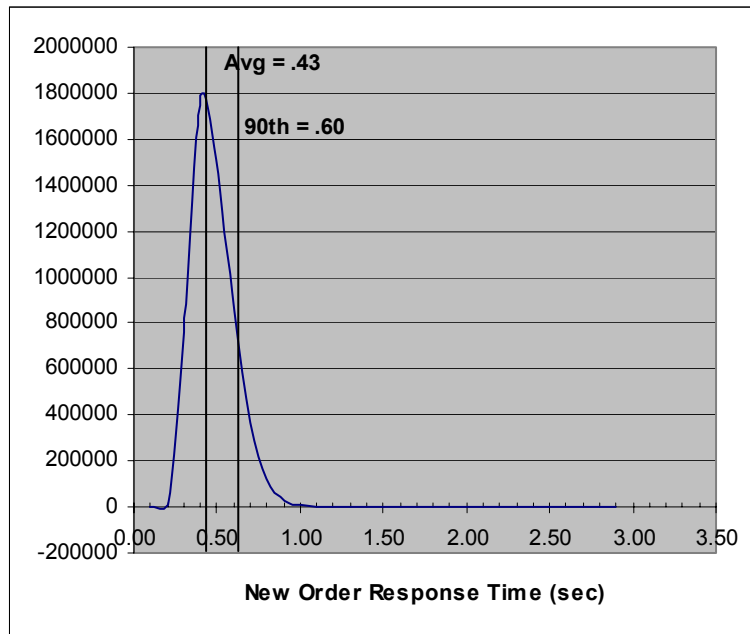


Figure 5-2. Payment Transaction - Response Time Frequency Distribution

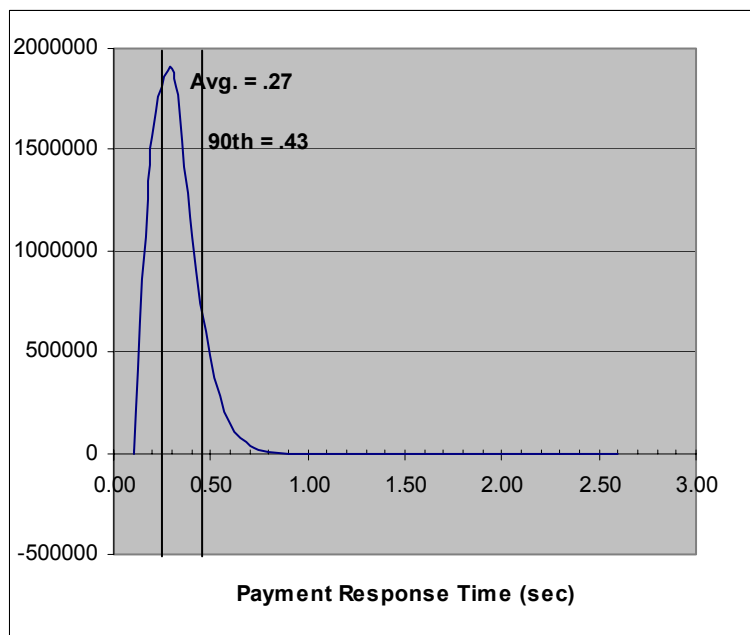


Figure 5-3. Order-Status Transaction - Response Time Frequency Distribution

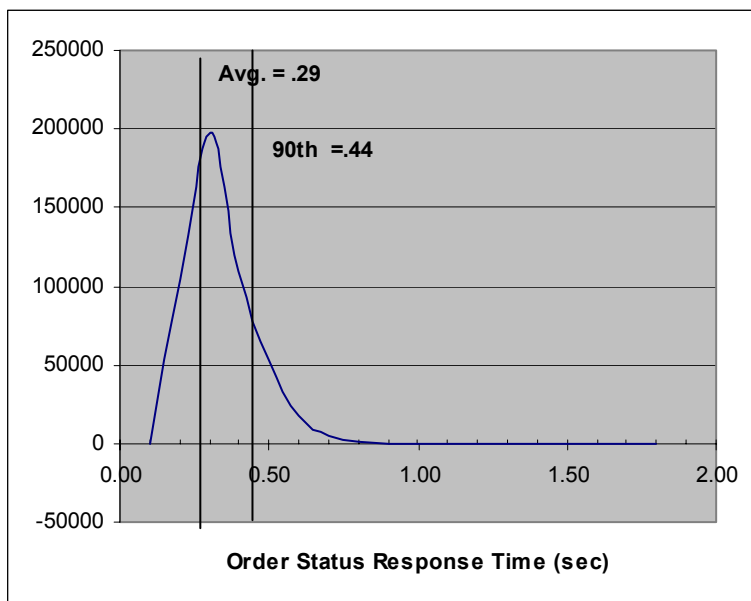
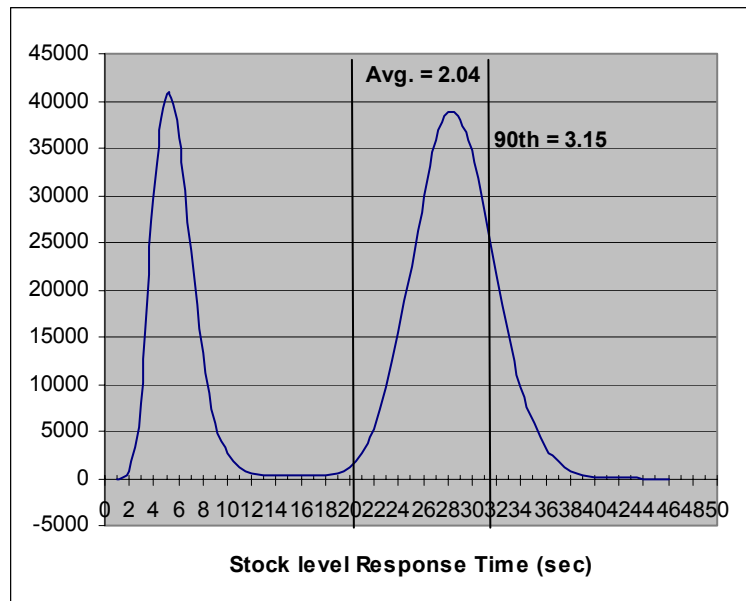


Figure 5-4. Delivery Transaction - Response Time Frequency Distribution



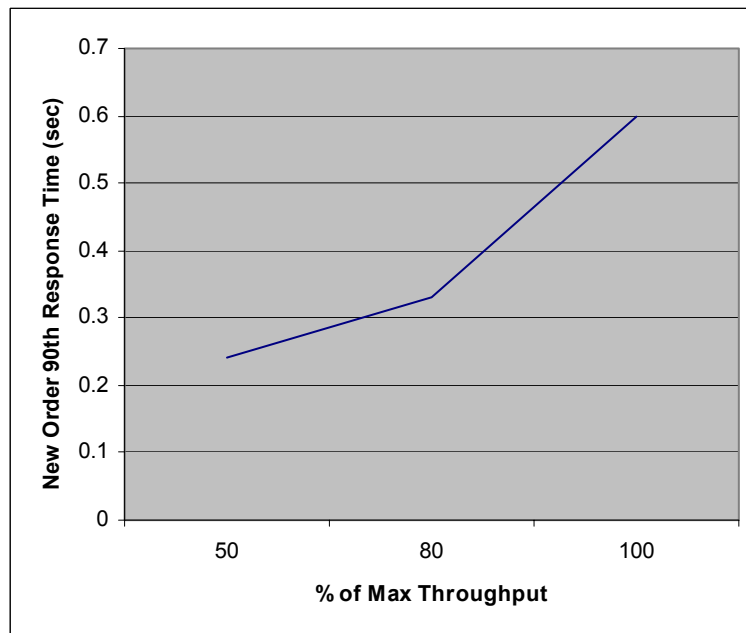
Figure 5-5. Stock-Level Transaction - Response Time Frequency Distribution



Performance Curve for Response Time vs. Throughput

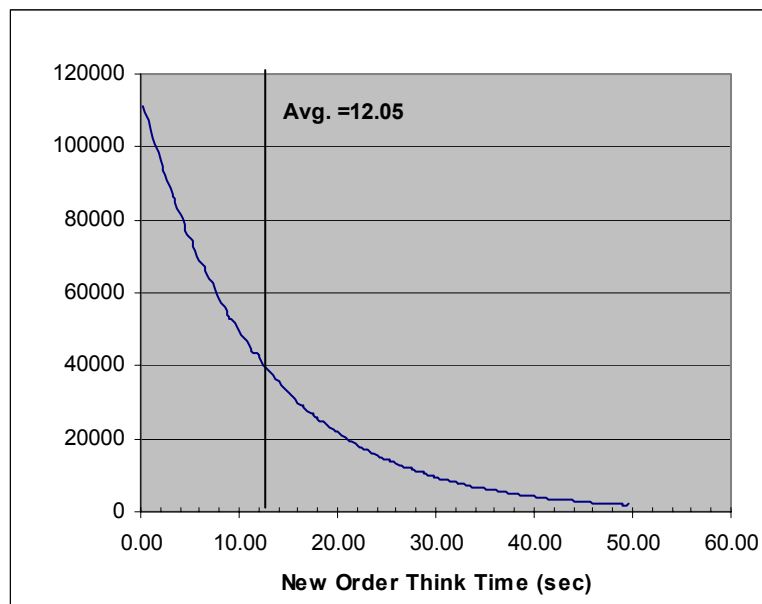
The performance curve for response time vs. throughput (see Clause 5.6.2) must be reported for the New-Order transaction.

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



New Order Think Time Distribution

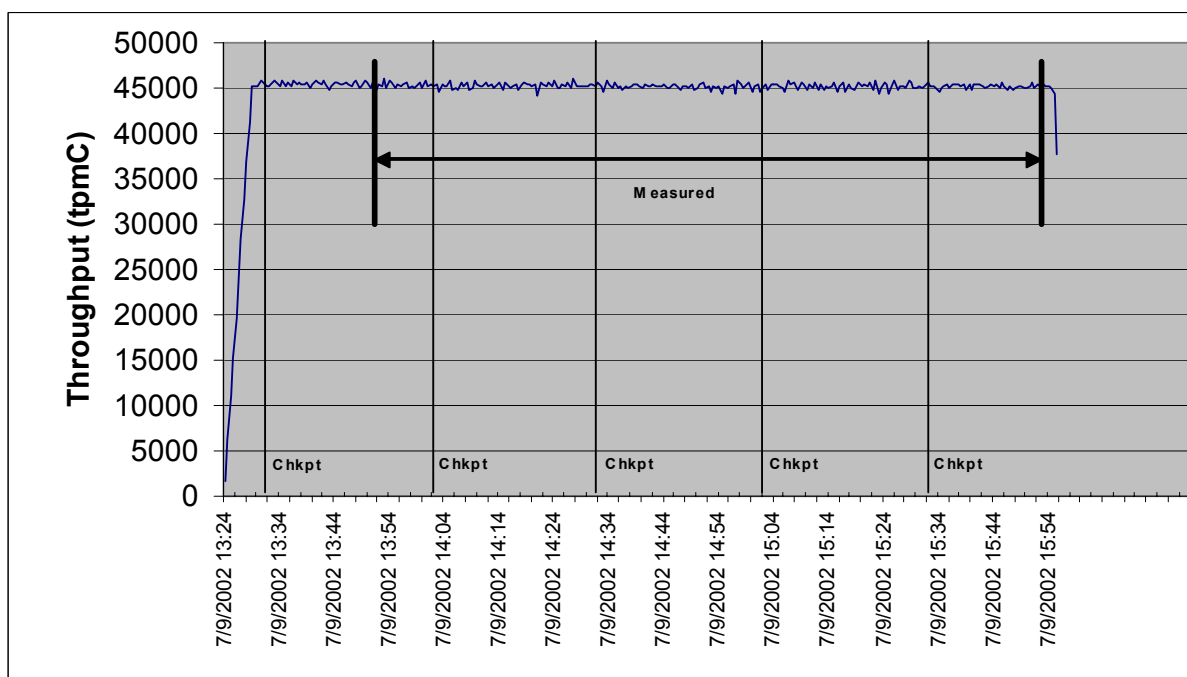
Figure 5-7. New-Order Think Time Distribution



Throughput vs. Elapsed Time

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

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



Steady State Methodology

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

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

Work Performed during Steady State

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

Transaction Flow

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

The RTE generated the required input data for the chosen transaction. It waited to complete the minimum required key time before transmitting the input screen. The transmission was time-stamped. The return of the screen with the required response data was time-stamped. The difference between these two time-stamps was the response time for that transaction and was logged in the RTE log. The RTE then waited the required think time interval before repeating the process starting at selecting another transaction from the menu.

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

Checkpoints

Checkpoints were executed on the server during the ramp-up phase and at 30-minute intervals. The measured run contained four checkpoints. SQL Server was started with trace flag 3502, which caused it to log the occurrence of the checkpoint. This information was used to verify that the checkpoints occurred at the appropriate times during the test run.

During a checkpoint, SQL Server flushes all dirty pages from its cache to disk. It places a record in the database transaction log indicating that the checkpoint has completed and that all transactions, which were committed prior to the checkpoint have been written to disk.

Measurement Interval

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

The measurement interval was 120 minutes.

Transaction Mix

The method of regulation of the transaction mix (e.g., card decks or weighted random distribution) must be described. If weighted distribution is used and the RTE adjusts the weights associated with each transaction type, the maximum adjustments to the weight from the initial value must be disclosed. (8.1.6.13)

See Table 5-3.

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

Percentage of Total Mix

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

See Table 5-3.

Table 5-3. Transaction Statistics and Transaction Mix

New Order	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	85.01
Remote warehouse payment transactions	14.99
Non-Primary Key Access	
Payment transactions using C_LAST	59.98
Order-Status transactions using C_LAST	60.06
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.87
Payment	43.04
Delivery	4.04
Stock Level	4.02
Order-Status	4.02

Number of Checkpoints

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

Checkpoints were performed during the ramp-up period and during each measured run interval. The first measurement interval checkpoint started 8 minutes and 37 seconds after the start of the measurement interval. The four checkpoints in the measured interval are shown in Table 5-4.

Table 5-4. Checkpoint Start Time and Duration

Checkpoint	Start Time	Duration
1	2:01:37 p.m.	13 minutes 30 seconds
2	2:31:35 p.m.	13 minutes 30 seconds
3	3:01:33 p.m.	13 minutes 30 seconds
4	3:31:31 p.m.	13 minutes 30 seconds

The checkpoints were verified to be clear of the protected zones around the beginning and end of the measurement intervals. The checkpoint interval was 30 minutes.

Clause 6: SUT, Driver and Communication Definition Related Items

Description of RTE

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

The RTE used was Microsoft BenchCraft 1.1.2 RTE. Benchcraft is a proprietary tool provided by Microsoft and is not commercially available. The RTE input is listed in Appendix C.

Emulated Components

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

No components were emulated.

Benchmarked and Targeted System Configuration Diagrams

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

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

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

Network Configuration

The network configurations of both the tested services and the proposed (target) services which are being represented and a thorough explanation of exactly which parts of the proposed configuration are being replaced with the Driver System must be disclosed (see Clause 6.6.4).

See the measured and priced configuration diagrams (pages 13 and 14) for details about the network configuration.

Network Bandwidth

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

The Ethernet used in the LAN complies with the IEEE.802.3 standard. The LANs that connected the driver RTEs to the clients had a bandwidth of 10Mbps. The LAN that connected the client to the server had a bandwidth of 100Mbps.

Operator Intervention

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

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

Clause 7: Pricing Related Items

Hardware and Software Components

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

Pricing source(s) and effective date(s) must also be reported.

The total 3-year price of the entire configuration must be reported, including: hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

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

Availability Date

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

The xSeries 360 is planned to be generally available on August 30, 2002. All other hardware and the software used in this benchmark are currently available.

Measured tpmC

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

- ◆ Maximum Qualified Throughput: 45,230.03 tpmC
- ◆ Price per tpmC: \$4.52 per tpmC
- ◆ Three-year cost of ownership: \$204,271

Country-Specific Pricing

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

The configuration is priced for the United States of America.

Usage Pricing

For any usage pricing, the sponsor must disclose:

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

The component pricing based on usage is shown below:

- 1 Microsoft Windows 2000 Advanced Server
- 2 Microsoft Windows 2000 Server
- 4 Microsoft SQL Server 2000 Enterprise Edition (based on per-processor price)

- 3-year support for hardware components (except for components for which 10 percent spares are provided)

System Pricing

System pricing should include subtotals for the following components: Server Hardware, Server Software, Client Hardware, Client Software, and Network Components used for terminal connection (see Clause 7.2.2.3).

System pricing must include line item indication where non-sponsoring companies' brands are used. System pricing must also include line item indication of third-party pricing.

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

Clause 9: Audit Related Items

Auditor

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

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

Availability of the Full Disclosure Report

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

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

InfoSizing, Inc.
1373 North Franklin Street
Colorado Springs, CO 80903
Phone: 719-473-7555
Fax: 719-473-7554

Benchmark Sponsors: William D. Hall
 Mgr., Server Systems Performance
 IBM Server Group
 3039 Cornwallis Road
 Research Triangle Park, NC 27709

July 17, 2002

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

Platform: IBM @server xSeries 360 c/s
 Operating system: Microsoft Windows 2000 Advanced Server
 Database Manager: Microsoft SQL Server 2000 Enterprise Edition
 Transaction Manager: Microsoft COM+ Transaction Manager

The results were:

CPU's Speed	Memory	Disks	NewOrder 90% Response Time	tpmC
Server: IBM @server xSeries 360				
4 x Xeon (1.6 GHz)	8 GB (1 MB cache/cpu)	169 x 18.2 GB 10Krpm	0.60 Seconds	45,230.03
Two Clients: IBM @server xSeries 220 (Specification for each)				
2 x Pentium III (1.4 GHz)	640 MB (512 KB cache/cpu)	1 x 18.2 GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC requirements for Revision 5.0 of the benchmark.

The following verification items were given special attention:

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

Additional Audit Notes:

None.

Respectfully Yours,

A handwritten signature in black ink, appearing to read 'François Raab', with a stylized flourish at the end.

François Raab, President

Appendix A: Source Code

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_time;
    char time_str[30];
```

```
char line_prefix[50];
va_list ap;

va_start(ap, format);

cur_time = time(&cur_time);
strftime(time_str, 29, "%X", localtime(&cur_time));

get_prefix(line_prefix);

fprintf(ERROROUT, "%s %s - ", line_prefix, time_str);
vfprintf(ERROROUT, format, ap);
fflush(ERROROUT);

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);
}

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/tpc.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 */

```

dlldata.c

```

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

DO NOT ALTER THIS FILE

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

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

*****/

#include <rpcproxy.h>

#ifdef __cplusplus
extern "C" {
#endif

EXTERN_PROXY_FILE( tpcc_com_ps )

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

DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )

```

```

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

/* end of generated dlldata file */

error.h

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

#pragma once

#ifdef _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;

#define ERR_FATAL_LEVEL 1
#define ERR_WARNING_LEVEL 2
#define ERR_INFORMATION_LEVEL 3

#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

```

```

class CBaseErr
{
public:
    char    *m_szApp;
    char    *m_szMsg;
    char    *m_szLoc; // code location where the error occurred
    int     m_idMsg;

    CBaseErr(void)
    {
        m_idMsg      = 0;
        m_szMsg      = new char[m_szMsg_size];
        m_szApp      = new char[m_szApp_size];
        m_szLoc      = NULL;

        m_szMsg[0]    = 0;
        m_szApp[0]    = 0;

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

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

    CBaseErr(int idMsg)

```

```

    {
        m_idMsg      = idMsg;
        m_szApp      = new char[m_szApp_size];
        m_szMsg      = new char[m_szMsg_size];
        m_szLoc      = NULL;

        GetModuleFileName(GetModuleHandle(NULL),
m_szApp, m_szApp_size);
        LoadString(GetModuleHandle(NULL), idMsg,
m_szMsg, m_szMsg_size);
    }

    CBaseErr(LPCTSTR szMsg)
    {
        m_idMsg      = 0;
        m_szApp      = new char[m_szApp_size];
        m_szMsg      = new char[m_szMsg_size];
        m_szLoc      = NULL;

        GetModuleFileName(GetModuleHandle(NULL),
m_szApp, m_szApp_size);
        strcpy(m_szMsg, szMsg);
    }

    void SetError(char *szMsg, LPCTSTR szLocation)
    {
        if (szMsg != NULL)
            strcpy(m_szMsg, szMsg);
        else
            m_szMsg[0] = 0;

        if (szLocation != NULL)
        {
            delete [] m_szLoc;
            m_szLoc = new char[strlen(szLocation)+1];
            strcpy(m_szLoc, szLocation);
        }
        else
        {
            delete [] m_szLoc;
            m_szLoc = NULL;
        }
    }

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

        if (szStr)
            j = wsprintf(szTmp, "%s\n", szStr);
        if (m_szLoc)
            j += wsprintf(szTmp+j,
"Location=%s\n", m_szLoc);
        if (m_szMsg)
            j += wsprintf(szTmp+j, "%s\n", m_szMsg);

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

    char *GetApp(void) { return m_szApp; }
    char *GetMsg(void) { return m_szMsg; }
    char *GetLocation(void) { return m_szLoc; }

    virtual int ErrorType() = 0; // a value which distinguishes the
kind of error that occurred
    virtual int ErrorNum() = 0; // an error value
specific to the error type

```

```

        virtual char *ErrorText() = 0;    // a string (i.e., human readable)
        representation of the error

```

```
};
```

```
class CSocketErr : public CBaseErr
```

```
{
public:
```

```
    enum Action
```

```
    {
```

```
        eNone,
        eSend,
        eSocket,
        eConnect
    };

```

```
    CSocketErr(Action eAction, LPCTSTR szLocation);
```

```
    CSocketErr(int iError) { m_errId = iError; };
```

```
    int m_errId;
```

```
    Action m_eAction;
```

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

```
    int ErrorNum() { return m_errId;};
```

```
    char *ErrorText(void);
```

```
};
```

```
class CSystemErr : public CBaseErr
```

```
{
public:
```

```
    enum Action
```

```
    {
```

```
        eNone,
        eTransactNamedPipe,
        eWaitNamedPipe,
        eSetNamedPipeHandleState,
        eCreateFile,
        eCreateProcess,
        eCallNamedPipe,
        eCreateEvent,
        eCreateThread,
        eVirtualAlloc,
        eReadFile,
        eWriteFile,
        eMapViewOfFile,
        eCreateFileMapping,
        eInitializeSecurityDescriptor,
        eSetSecurityDescriptorDacl,
        eCreateNamedPipe,
        eConnectNamedPipe,
        eWaitForSingleObject,
        eRegOpenKeyEx,
        eRegQueryValueEx,
    };

```

```
    CSystemErr(Action eAction, LPCTSTR szLocation);
```

```
    void Draw(HWND hwnd, LPCTSTR szStr = NULL);
```

```
    int m_errId;
```

```
    Action m_eAction;
```

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

```
    int ErrorNum() { return m_errId;}
```

```
    char *ErrorText() { return m_szMsg; }
```

```
};
```

```
class CMemoryErr : public CBaseErr
```

```
{
```

```
public:
```

```
    CMemoryErr(void);
```

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

```
    int ErrorNum() { return 0;}
```

```
    char *ErrorText() { return "Insufficient Memory to continue.";}
```

```
};
```

install.c

```

/*      FILE:          INSTALL.C
 *
 *      Microsoft TPC-C Kit Ver.
 *
 *      4.20.000
 *
 *      Copyright Microsoft, 1999
 *
 *      All Rights Reserved
 *
 *      not audited
 *
 *      PURPOSE:        Automated installation application for TPC-C
 *      Web Kit
 *      Contact:   Charles Levine (clevine@microsoft.com)
 *
 *      Change history:
 *
 *      4.20.000 - added COM installation steps
 */

```

```
#include <windows.h>
```

```
#include <direct.h>
```

```
#include <io.h>
```

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
#include <commctrl.h>
```

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

```
#include "resource.h"
```

```
#define WM_INITTEXT WM_USER+100
```

```
HICON hIcon;
```

```
HINSTANCE hInst;
```

```
DWORD versionExeMS;
```

```
DWORD versionExeLS;
```

```
DWORD versionExeMM;
```

```
DWORD versionDllMS;
```

```
DWORD versionDllLS;
```

```
// TPC-C registry settings
```

```
TPCCREGISTRYDATA Reg;
```

```
static int iPoolThreadLimit;
```

```
static int iThreadTimeout;
```

```
static int iListenBackLog;
```

```
static int iAcceptExOutstanding;
```

```
static int iMaxPhysicalMemory;
```

```
//max physical memory in MB
```

```
static char szLastFileName[64]; // last file we worked
on (for error reporting)
```

```

BOOL CALLBACK LicenseDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam);

```

```

BOOL CALLBACK UpdatedDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam);

```

```

BOOL CALLBACK MainDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam);

```

```

BOOL CALLBACK CopyDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam);
static void ProcessOK(HWND hwnd, char *szDllPath);
static void ReadRegistrySettings(void);
static void WriteRegistrySettings(char *szDllPath);
static BOOL RegisterDLL(char *szFileName);
static int CopyFiles(HWND hDlg, char
*szDllPath);
static BOOL GetInstallPath(char *szDllPath);
static void GetVersionInfo(char *szDllPath, char
*szExePath);
static BOOL CheckWWWService(void);
static BOOL StartWWWService(void);
static BOOL StopWWWService(void);
static void UpdateDialog(HWND hDlg);

BOOL install_com(char *szDllPath);

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

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

    hInst = hInstance;

    InitCommonControls();

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

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

    DestroyIcon(hIcon);
    return 0;
}

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

    switch(uMsg)
    {
        case WM_INITDIALOG:
            hFont = CreateFont(-12, 0, 0, 0, 400, 0, 0, 0,
0, 0, 0, 0, "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;

```

<pre> MEMORYSTATUS memoryStatus; OSVERSIONINFO VI; char szTmp[256]; static char szDllPath[256]; static char szExePath[256]; switch(uMsg) { case WM_INITDIALOG: GlobalMemoryStatus(&memoryStatus); iMaxPhysicalMemory= (memoryStatus.dwTotalPhys/ 1048576); if (GetInstallPath(szDllPath)) { MessageBox(hwnd, "Error internet service inetsrv is not installed.", NULL, MB_ICONSTOP MB_OK); EndDialog(hwnd, FALSE); return TRUE; } // set default values ZeroMemory(&Reg, sizeof(Reg)); Reg.dwNumberOfDeliveryThreads = 4; Reg.dwMaxConnections = 100; Reg.dwMaxPendingDeliveries = 100; Reg.eDB_Protocol = DBLIB; Reg.eTxnMon = None; strcpy(Reg.szDbServer, ""); strcpy(Reg.szDbName, "tpcc"); strcpy(Reg.szDbUser, "sa"); strcpy(Reg.szDbPassword, ""); iPoolThreadLimit = iMaxPhysicalMemory * 2; iThreadTimeout = 86400; iListenBackLog = 15; iAcceptExOutstanding = 40; ReadTPCCRegistrySettings(&Reg); ReadRegistrySettings(); GetModuleFileName(hInst, szExePath, sizeof(szExePath)); GetVersionInfo(szDllPath, szExePath); wsprintf(szTmp, "Version %d.%2.2d.%3.3d", versionExeMS, versionExeMM, versionExeLS); SetDlgItemText(hwnd, IDC_VERSION, szTmp); SetDlgItemText(hwnd, IDC_PATH, szDllPath); SetDlgItemText(hwnd, ED_DB_SERVER, Reg.szDbServer); SetDlgItemText(hwnd, ED_DB_USER_ID, Reg.szDbUser); SetDlgItemText(hwnd, ED_DB_PASSWORD, Reg.szDbPassword); SetDlgItemText(hwnd, ED_DB_NAME, Reg.szDbName); SetDlgItemInt(hwnd, ED_THREADS, Reg.dwNumberOfDeliveryThreads, FALSE); </pre>	<pre> 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)) { </pre>
--	---

<pre> 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: return ProcessOK(hwnd, szDllPath); case IDCANCEL: return EndDialog(hwnd, FALSE); default: return FALSE; } break; default: break; } return FALSE; } static void ProcessOK(HWND hwnd, char *szDllPath) { int d; HWND hDlg; int rc; char szFullName[256]; char szErrMsg[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; </pre>	<pre> 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); // write binaries to inetpub\wwwroot rc = CopyFiles(hDlg, szDllPath); if (!rc) { ShowWindow(hwnd, SW_SHOWNA); DestroyWindow(hDlg); strcpy(szErrMsg, "Error(s) occurred when creating "); strcat(szErrMsg, szLastFileName); MessageBox(hwnd, szErrMsg, NULL, MB_ICONSTOP MB_OK); EndDialog(hwnd, 0); return; } // update registry SetDlgItemText(hDlg, IDC_STATUS, "Updating Registry."); SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0); UpdateDialog(hDlg); WriteRegistrySettings(szDllPath); // register com proxy stub strcpy(szFullName, szDllPath); strcat(szFullName, "tpcc_com_ps.dll"); if (!RegisterDLL(szFullName)) { ShowWindow(hwnd, SW_SHOWNA); DestroyWindow(hDlg); strcpy(szErrMsg, "Error occurred when registering "); strcat(szErrMsg, szFullName); MessageBox(hwnd, szErrMsg, NULL, MB_ICONSTOP MB_OK); EndDialog(hwnd, 0); return; </pre>
--	--

```

    }

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

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

        Sleep(100);

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

        EndDialog(hwnd, rc);
        return;
    }

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

    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Parameters", 0,
KEY_READ, &hKey) == ERROR_SUCCESS )
    {
        size = sizeof(iPoolThreadLimit);
        if ( RegQueryValueEx(hKey, "PoolThreadLimit", 0,
&type, (char *)&iPoolThreadLimit, &size) == ERROR_SUCCESS )
            if ( !iPoolThreadLimit )
                iPoolThreadLimit =
iMaxPhysicalMemory * 2;

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

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

        RegCloseKey(hKey);
    }

    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters", 0,
KEY_READ, &hKey) == ERROR_SUCCESS )

```

```

    {
        size = sizeof(iAcceptExOutstanding);
        if ( RegQueryValueEx(hKey, "AcceptExOutstanding", 0,
&type, (char *)&iAcceptExOutstanding, &size) == ERROR_SUCCESS )
            if ( !iAcceptExOutstanding )
                iAcceptExOutstanding = 40;

        RegCloseKey(hKey);
    }
}

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

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

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

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

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

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

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

        RegFlushKey(hKey);
        RegCloseKey(hKey);
    }

    if ( (iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Parameters", 0, NULL,
REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, NULL, &hKey,
&dwDisposition)) == ERROR_SUCCESS )

```

```

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

        RegFlushKey(hKey);
        RegCloseKey(hKey);
    }

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

        RegFlushKey(hKey);
        RegCloseKey(hKey);
    }

    return;
}

BOOL CALLBACK CopyDlgProc(HWND hwnd, UINT uMsg, WPARAM
wParam, LPARAM lParam)
{
    if ( uMsg == WM_INITDIALOG )
    {
        SendDlgItemMessage(hwnd, IDC_PROGRESS1,
PBM_SETRANGE, 0, MAKELPARAM(0, 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)
{
    BOOL            bSvcRunning;

    bSvcRunning = CheckWWWebService();
    if ( bSvcRunning )
    {
        SetDlgItemText(hDlg, IDC_STATUS, "Stopping Web
Service.");
        SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

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

    SetDlgItemText(hDlg, IDC_STATUS, "Copying Files...");
    SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0,
0);
    UpdateDialog(hDlg);

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

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

```

```

        UpdateDialog(hDlg);

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

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

        // install tpcc_tuxedo.dll
        strcpy( szLastFileName, "tpcc_tuxedo.dll" );
        if (!FileFromResource( "TUXEDO_DLL", IDR_TUXEDO_DLL,
szDllPath, szLastFileName ))
            return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0,
0);
        UpdateDialog(hDlg);

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

        // install tpcc_com_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);

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

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

        return 1;
    }

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

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

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

        RegCloseKey(hKey);
    }

    return bRc;
}

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

    versionDllMS = 0;
    versionDllLS = 0;

```

```

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

        VerQueryValue(ptr, "\\", &vs, &dwBytes);
        versionDllMS = vs->dwProductVersionMS;
        versionDllLS = vs->dwProductVersionLS;
        free(ptr);
    }

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

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

static BOOL 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 StartWWWWebErr;
    //start Service pending, Check the status until the service is running.
    if (! QueryServiceStatus(schService, &ssStatus) )
        goto StartWWWWebErr;
    while( ssStatus.dwCurrentState != SERVICE_RUNNING)
    {

        dwOldCheckPoint = ssStatus.dwCheckPoint;
        //Save the current checkpoint.
        Sleep(ssStatus.dwWaitHint);
        //Wait for the specified interval.
        if ( !QueryServiceStatus(schService, &ssStatus) )
            //Check the status again.
            break;

        if (dwOldCheckPoint >= ssStatus.dwCheckPoint)
            //Break if the checkpoint has not been incremented.
            break;
    }

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

    CloseServiceHandle(schService);
    return TRUE;

StartWWWWebErr:
    CloseServiceHandle(schService);
    return FALSE;
}

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

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

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

    if ( !ControlService(schService, SERVICE_CONTROL_STOP,
&ssStatus) )
        goto StopWWWWebErr;
    //start Service pending, Check the status until the service is running.
    if (! QueryServiceStatus(schService, &ssStatus) )
        goto StopWWWWebErr;
    while( ssStatus.dwCurrentState == SERVICE_RUNNING)
    {

```

```

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

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

    CloseServiceHandle(schService);
    return TRUE;

StopWWWebErr:
    CloseServiceHandle(schService);
    return FALSE;
}

static void UpdateDialog(HWND hDlg)
{
    MSG msg;

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

```

install.h

```

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

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

#define BN_LOG                      1001
#define ED_KEEP                     1002
#define ED_THREADS                  1003
#define ED_THREADS2                 1004
#define IDC_PATH                    1007
#define IDC_VERSION                 1009
#define IDC_RESULTS                 1010
#define IDC_PROGRESS1              1011
#define IDC_STATUS                  1012
#define IDC_BUTTON1                 1013
#define ED_MAXCONNECTION           1014
#define ED_IIS_MAX_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 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)
#ifndef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif // _WIN32

//
// Dialog
//

IDD_DIALOG1 DIALOGEX 0, 0, 219, 351
STYLE DS_MODALFRAME | DS_CENTER | WS_MINIMIZEBOX |
WS_POPUP | WS_CAPTION |
WS_SYSMENU
CAPTION "TPC-C Web Client Installation Utility"
FONT 8, "MS Sans Serif"
BEGIN
    EDITTEXT    ED_THREADS,164,45,34,12,ES_RIGHT | ES_NUMBER,
                WS_EX_RTLREADING
    EDITTEXT    ED_MAXDELIVERIES,164,59,34,12,ES_RIGHT |
ES_NUMBER,
                WS_EX_RTLREADING
    EDITTEXT    ED_MAXCONNECTION,164,73,34,12,ES_RIGHT |
ES_NUMBER,
                WS_EX_RTLREADING
    CONTROL
    "None",IDC_TM_NONE,"Button",BS_AUTORADIOBUTTON |
                WS_GROUP | WS_TABSTOP,43,100,33,10
    CONTROL
    "COM",IDC_TM_MTS,"Button",BS_AUTORADIOBUTTON |
                WS_TABSTOP,43,113,32,10
    CONTROL
    "TUXEDO",IDC_TM_TUXEDO,"Button",BS_AUTORADIOBUTTON |
                WS_TABSTOP,106,100,46,10
    CONTROL
    "ENCINA",IDC_TM_ENCINA,"Button",BS_AUTORADIOBUTTON |

```

```

        WS_DISABLED | WS_TABSTOP,106,113,43,10
EDITTEXT    ED_DB_SERVER,131,152,67,12,ES_AUTOHSCROLL
EDITTEXT    ED_DB_USER_ID,131,165,67,12,ES_AUTOHSCROLL
EDITTEXT    ED_DB_PASSWORD,131,178,67,12,ES_AUTOHSCROLL
EDITTEXT    ED_DB_NAME,131,191,67,12,ES_AUTOHSCROLL
CONTROL
"DBLIB",IDC_DBLIB,"Button",BS_AUTORADIOBUTTON | WS_GROUP |
    WS_TABSTOP,45,219,39,12
CONTROL
"ODBC",IDC_ODBC,"Button",BS_AUTORADIOBUTTON | WS_TABSTOP,
    91,219,39,12
EDITTEXT
ED_IIS_MAX_THREAD_POOL_LIMIT,164,263,34,12,ES_RIGHT |
    ES_NUMBER,WS_EX_RTREADING
EDITTEXT
ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,164,277,34,12,ES_RIGHT |
    ES_NUMBER,WS_EX_RTREADING
EDITTEXT    ED_IIS_THREAD_TIMEOUT,164,291,34,12,ES_RIGHT |
ES_NUMBER,
    WS_EX_RTREADING
EDITTEXT    ED_IIS_LISTEN_BACKLOG,164,305,34,12,ES_RIGHT |
ES_NUMBER,
    WS_EX_RTREADING
DEFPUSHBUTTON "OK",IDOK,53,331,50,14
PUSHBUTTON   "Cancel",IDCANCEL,119,331,50,14
EDITTEXT    IDC_PATH,106,26,91,13,ES_AUTOHSCROLL |
ES_READONLY
LTEXT       "Number of Delivery Threads:",IDC_STATIC,35,45,115,12
LTEXT       "Max Number of Connections:",IDC_STATIC,35,73,115,12
RTEXT       "Version 4.11",IDC_VERSION,120,4,89,9
LTEXT       "IIS Max Thread Pool Limit:",IDC_STATIC,36,263,115,12
LTEXT       "Web Service Backlog Queue
Size:",IDC_STATIC,36,277,115,
    12
LTEXT       "IIS Thread Timeout (seconds):",IDC_STATIC,36,291,115,12
LTEXT       "IIS Listen Backlog:",IDC_STATIC,36,307,115,10
GROUPBOX    "Database
Interface",IDC_STATIC,35,208,163,27,WS_GROUP
LTEXT       "Installation directory:",IDC_STATIC,35,29,71,10
GROUPBOX    "Transaction Monitor",IDC_STATIC,33,90,165,37
LTEXT       "Server Name:",IDC_STATIC,35,155,56,8
LTEXT       "User ID:",IDC_STATIC,35,168,60,8
LTEXT       "User Password:",IDC_STATIC,35,181,83,8
LTEXT       "Database Name:",IDC_STATIC,35,194,54,8
GROUPBOX    "SQL Server Connection
Properties",IDC_STATIC,22,139,187,
    102
GROUPBOX    "Web Client Properties",IDC_STATIC,22,15,187,118
GROUPBOX    "IIS Settings",IDC_STATIC,22,247,187,79
LTEXT       "Max Pending Deliveries:",IDC_STATIC,35,59,115,12
END

IDD_DIALOG2 DIALOGEX 0, 0, 117, 62
STYLE DS_SETFOREGROUND | DS_3DLOOK | DS_CENTER |
WS_POPUP | WS_BORDER
EXSTYLE WS_EX_STATICEDGE
FONT 12, "MS Sans Serif", 0, 0, 0x1
BEGIN
    DEFPUSHBUTTON "OK",IDOK,33,45,50,9
    CTEXT        "HTML TPC-C Installation
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 DISCARDABLE 0, 0, 91, 40
STYLE DS_SYSMODAL | DS_MODALFRAME | DS_3DLOOK |
DS_CENTER | WS_CAPTION
CAPTION "Installing TPC-C Web Client"
FONT 12, "Arial Black"
BEGIN
    CONTROL
        "Progress1",IDC_PROGRESS1,"msctls_progress32",WS_BORDER,
            7,20,77,13
    CTEXT      "Static",IDC_STATUS,7,7,77,12,SS_SUNKEN
END

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

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

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

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

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

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

```

```

#endif // APSTUDIO_INVOKED

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

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

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

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

#endif // APSTUDIO_INVOKED

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

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

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

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

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

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

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

IDR_LICENSE1     LICENSE DISCARDABLE "license.txt"

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

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

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

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

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

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

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

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

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

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

```

```

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

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

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

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

////////////////////////////////////
//
// COM_TYPLIB
//

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

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

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

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

install_com.cpp

/*      FILE:          INSTALL_COM.CPP
 *
 *      4.20.000      Microsoft TPC-C Kit Ver.
 *
 *      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"))
        {

```

<pre> 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; </pre>	<pre> 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; </pre>
--	---

```

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

        // iterate through interfaces in component
        while (ICountItf > 0)
        {
            hr =
pCatalogCollectionItf->get_Item(ICountItf - 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(&ICountMethod);
            if (!SUCCEEDED(hr)) goto Error;

            // iterate through methods of interface
            while (ICountMethod > 0)
            {
                hr =
pCatalogCollectionMethod->get_Item(ICountMethod - 1, (IDispatch**)
&pCatalogObjectMethod);

                if (!SUCCEEDED(hr)) goto Error;

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

                pCatalogObjectMethod->Release();

                pCatalogObjectMethod = NULL;

                ICountMethod--;

            }

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

            pCatalogObjectItf->Release();
            pCatalogObjectItf = NULL;

            ICountItf--;

        }

        pCatalogObjectCo->Release();
        pCatalogObjectCo = NULL;

```

```

        ICountCo--;
    }

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

    pCatalogCollectionApp->Release();
    pCatalogCollectionApp = NULL;

    pCatalogCollectionCo->Release();
    pCatalogCollectionCo = NULL;

    pCatalogCollectionItf->Release();
    pCatalogCollectionItf = NULL;

    pCatalogCollectionMethod->Release();
    pCatalogCollectionMethod = NULL;

Error:
    CoUninitialize();

    if (!SUCCEEDED(hr))
    {
        LPTSTR lpBuf;
        DWORD dwRes =
FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER |
FORMAT_MESSAGE_FROM_SYSTEM,

        NULL,

        hr,

        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),

        (LPTSTR) &lpBuf,

        0,

        NULL);

        // _tprintf(__T("Error adding components. HRESULT:
0x%x\n%s"), hr, lpBuf);

        return TRUE;

    }
    else
        return FALSE;

}

```

license.txt

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

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

The SOFTWARE PRODUCT is protected by copyright laws

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

1. GRANT OF LICENSE. This EULA grants you the following rights:

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

2. RESTRICTIONS.

--You must maintain all copyright notices on all copies of the SOFTWARE PRODUCT.

--You may not distribute copies of the SOFTWARE PRODUCT to third parties.

--You may not rent, lease or lend the SOFTWARE PRODUCT.

--You may not use the SOFTWARE PRODUCT or any derivative works thereof to internally test database management system software other than Microsoft SQL Server and/or operating system software other than Microsoft Windows NT.

-- You may not disclose the results of any benchmark tests using the SOFTWARE PRODUCT to any third party without Microsoft's prior written approval.

-- You may not disclose or provide the SOFTWARE PRODUCT or any derivative works thereof, or any information relating to the SOFTWARE PRODUCT (including the existence of the SOFTWARE PRODUCT or the results of use and testing or benchmark testing), to any third party without Microsoft's written permission.

3. TERMINATION. Without prejudice to any other rights, Microsoft may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE PRODUCT.

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

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

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

7. EXPORT RESTRICTIONS.

You agree that you will not export or re-export the SOFTWARE PRODUCT to any country, person, entity or end user subject to U.S.A. export restrictions. Restricted countries currently include, but are not necessarily limited to Cuba, Iran, Iraq, Libya, North Korea, Syria, and the Federal Republic of Yugoslavia (Serbia and Montenegro, U.N. Protected Areas and areas of Republic of Bosnia and Herzegovina under the control of Bosnian Serb forces). You warrant and represent that neither the U.S.A. Bureau of Export Administration nor any other federal agency has suspended, revoked or denied your export privileges.

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

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

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

11. MISCELLANEOUS

This EULA is governed by the laws of the State of Washington, U.S.A.

Should you have any questions concerning this EULA, or if you desire to contact Microsoft for any reason, please contact the Microsoft subsidiary serving your country, or write:

Microsoft Sales Information Center/One Microsoft Way/Redmond, WA 98052-6399.

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

EXCLUSION DE GARANTIES. Microsoft renonce entièrement ... toute garantie pour le LOGICIEL. Le LOGICIEL et toute autre documentation s'y rapportant sont fournis ® comme tels - sans aucune garantie quelle qu'elle soit, expresse ou implicite, y compris, mais ne se limitant pas aux garanties implicites de la qualité, marchande ou un usage particulier. Le risque total d'écoulement de l'utilisation ou de la

performance du LOGICIEL est entre vos mains.

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

ABSENCE DE RESPONSABILITÉ POUR LES DOMMAGES INDIRECTS.

Microsoft ou ses fournisseurs ne pourront être tenus responsables en aucune circonstance de tout dommage quel qu'il soit (y compris mais non de façon limitative les dommages directs ou indirects causés par la perte de données, l'interruption des affaires, la perte d'information commerciale ou toute autre perte pécuniaire) résultant de l'utilisation ou de l'impossibilité d'utilisation de ce produit, et ce, même si la société, Microsoft a, à l'avance, avisé de l'éventualité de tels dommages. Certains États/juridictions ne permettent pas l'exclusion ou la limitation de responsabilité relative aux dommages indirects ou consécutifs, et la limitation ci-dessus peut ne pas s'appliquer à votre juridiction. La présente Convention est régie par les lois de la province d'Ontario, Canada. Chacune des parties ... la présente reconnaît irrévocablement la compétence des tribunaux de la province d'Ontario et consent ... instituer tout litige qui pourrait découler de la présente auprès des tribunaux situés dans le district judiciaire de York, province d'Ontario. Au cas où vous auriez des questions concernant cette licence ou que vous désiriez vous mettre en rapport avec Microsoft pour quelque raison que ce soit, veuillez contacter la succursale Microsoft desservant votre pays, dont l'adresse est fournie dans ce produit, ou écrire ... : Microsoft Customer Sales and Service, One Microsoft Way, Redmond, Washington 98052 6399.

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); \
    \
    (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;
        DPRINT(("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;
        DPRINT(("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),sta
    tus);
    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\\TxTpc";
    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
 */
#ifdef __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();
#ifdef __cplusplus
}
#endif

#endif /* MON_CLIENT_H */

```

readme.txt

ReadRegistry.cpp

```

/*      FILE:          READREGISTRY.CPP
 *
 *      Microsoft TPC-C Kit Ver.
 *
 *      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;

        RegCloseKey(hKey);

        return FALSE;
    }

```

ReadRegistry.h

```

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

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

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

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

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );

```

Resource.h

```

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

```

```

#define IDD_DIALOG3          108
#define IDR_LICENSE1         112
#define IDD_DIALOG4          113
#define IDR_TPCCOBJ1         117
#define IDR_TPCCSTUB1        118
#define IDR_DBLIB_DLL        122
#define IDR_ODBC_DLL         123
#define IDR_TUXEDO_APP       124
#define IDR_TUXEDO_DLL       125
#define IDR_COM_DLL          126
#define IDR_COMPS_DLL        127
#define IDR_COMALL_DLL       128
#define IDR_COMTYPLIB_DLL    129
#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
#ifdef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE 130
#define _APS_NEXT_COMMAND_VALUE 40001
#define _APS_NEXT_CONTROL_VALUE 1031
#define _APS_NEXT_SYMED_VALUE 101
#endif
#endif

```

Resource_tpcc_rc.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

```

rtetime.h

```

/* FILE: rtetime.h : header file
 * Copyright 1997 Microsoft Corp., All rights reserved.
 *
 * Authors: Charles Levine, Philip Durr
 *
 * Microsoft Corp.
 */

#define MAX_JULIAN_TIME
0x7FFFFFFFFFFFFFFF
#define JULIAN_TIME          __int64
#define TC_TIME              DWORD
extern "C"
{
    BOOL          InitJulianTime(LPSYSTEMTIME lpInitTime);
    JULIAN_TIME   GetJulianTime(void);
    DWORD         MyTickCount(void);
    void          GetJulianAndTC(JULIAN_TIME *pJulian, DWORD
    *pTC);
    JULIAN_TIME   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.
 *
 * Authors: Mike Parkes, Charles Levine, Philip Durr
 *
 * Microsoft Corp.
 */

#ifdef _INC_Spinlock

    const LONG LockClosed      = 1;
    const LONG LockOpen        = 0;

/*****
 *
 * Spinlock and Semaphore locking.
 *
 * This class provides a very conservative locking scheme.
 * The assumption behind the code is that locks will be
 * held for a very short time. When a lock is taken a memory
 * location is exchanged. All other threads that want this
 * lock wait by spinning and sometimes sleeping on a semaphore
 * until it becomes free again. The only other choice is not
 * to wait at all and move on to do something else. This
 * module should normally be used in conjunction with cache
 * aligned memory in minimize cache line misses.
 *
 */

```

```

*****/

class Spinlock
{
    // Private data.
    HANDLE      Semaphore;
    volatile LONG m_Spinlock;
    volatile LONG Waiting;

#ifdef _DEBUG
    // Counters for debugging 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 );

};

*****/

/*
 * 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

tpcc.cpp

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

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

#include <sqltypes.h>

```

```

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

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

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

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

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

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

#define LEN_ERR_STRING 256

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

char
szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

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

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

static CRITICAL_SECTION TermCriticalSection;

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

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

// For deferred Delivery txns:
CTxnLog //used to log delivery transaction information *txnDelilog = NULL;

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

// configuration settings from registry
TPCCREGISTRYDATA Reg;

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

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

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

```

<pre> 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"); 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"); LoadLibrary(szDllName); if (hLibInstanceTm == NULL) throw new CWEBCLNT_ERR(ERR_LOADDLL_FAILED, szDllName, GetLastError()); // get function pointer to wrapper for class constructor </pre>	<pre> 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()); } </pre>
--	--

<pre> else if (Reg.eDB_Protocol == ODBC) { strcpy(szDllName, Reg.szPath); strcat(szDllName, "tpcc_odbc.dll"); hLibInstanceDb = LoadLibrary(szDllName); if (hLibInstanceDb == NULL) throw new CWEBCLNT_ERR(ERR_LOADDLL_FAILED, szDllName, GetLastError()); // get function pointer to wrapper for class constructor pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*) GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new"); if (pCTPCC_ODBC_new == NULL) throw new CWEBCLNT_ERR(ERR_GETPROCADDR_FAILED, szDllName, GetLastError()); } if (dwNumDeliveryThreads) { // for deferred delivery txns: hDoneEvent = CreateEvent(NULL, TRUE /* manual reset */, FALSE /* initially not signalled */, NULL); InitializeCriticalSection(&DelBuffCriticalSection); hWorkerSemaphore = CreateSemaphore(NULL, 0, dwDelBuffSize, NULL); dwDelBuffFreeCount = dwDelBuffSize; InitJulianTime(NULL); // create unique log file name based on delilog-yyymmdd-hhmm.log SYSTEMTIME Time; GetLocalTime(&Time); wsprintf(szLogFile, "%sdelivery-%2.2d%2.2d%2.2d-%2.2d%2.2d.log", Reg.szPath, Time.wYear % 100, Time.wMonth, Time.wDay, Time.wHour, Time.wMinute); txndelilog = new CTxnLog(szLogFile, TXN_LOG_WRITE); //write event into txn log for START txndelilog->WriteCtrlRecToLog(TXN_EVENT_START, szMyComputerName, sizeof(szMyComputerName)); // allocate structures for delivery buffers and thread mgmt pDeliHandles = new HANDLE[dwNumDeliveryThreads]; </pre>	<pre> pDelBuff = new DELIVERY_TRANSACTION[dwDelBuffSize]; // launch DeliveryWorkerThread to perform actual delivery txns for(i=0; i<dwNumDeliveryThreads; i++) { pDeliHandles[i] = (HANDLE) _beginthread(DeliveryWorkerThread, 0, NULL); if (pDeliHandles[i] == INVALID_HANDLE_VALUE) throw new CWEBCLNT_ERR(ERR_DELIVERY_THREAD_FAILED); } break; case DLL_PROCESS_DETACH: if (dwNumDeliveryThreads) { if (txndelilog != NULL) { //write event into txn log for STOP txndelilog->WriteCtrlRecToLog(TXN_EVENT_STOP, szMyComputerName, sizeof(szMyComputerName)); // This will do a clean shutdown of the delivery log file CTxnLog *txndelilogLocal = txndelilog; txndelilog= NULL; delete txndelilogLocal; } delete [] pDeliHandles; delete [] pDelBuff; CloseHandle(hWorkerSemaphore); CloseHandle(hDoneEvent); DeleteCriticalSection(&DelBuffCriticalSection); DeleteCriticalSection(&TermCriticalSection); if (hLibInstanceTm != NULL) FreeLibrary(hLibInstanceTm); hLibInstanceTm = NULL; if (hLibInstanceDb != NULL) FreeLibrary(hLibInstanceDb); hLibInstanceDb = NULL; Sleep(500); break; </pre>
--	--

```

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

return TRUE;
}

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

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

```

```

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

        else if (Reg.eDB_Protocol == DBLIB)
            pTxn = pCTPCC_DBLIB_new(
Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, szMyComputerName,
Reg.szDbName );

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

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

            txnDeliLog->WriteToLog(&txnDeliRec);

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

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

ErrorExit:
    delete pTxn;
    _endthread();
}

/* FUNCTION: PostDeliveryInfo
*
* PURPOSE:      This function enters the delivery txn into the deferred
delivery buffer.
*

```

```

* RETURNS:      BOOL    FALSE    delivery information
posted successfully
*
*              TRUE
error cannot post delivery info
*/

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

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

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

    LeaveCriticalSection(&DelBuffCriticalSection);

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

    return bError;
}

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

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

    //allowable client command strings i.e. CMD=command
    static char *szCmds[] =
{

```

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

" " face=\"Courier New\"><PRE>" " " __DATE__ ", " __TIME__ "
" " " __FILE__ " (" __TIMESTAMP__ ")
" " "</PRE>" " ACTION=\"tpcc.dll\" METHOD=\"GET\">" " TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">" " TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">" </pre>	<pre> "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"1\">" "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"0\">" "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"0\">" "<INPUT TYPE=\"hidden\" NAME=\"VERSION\" VALUE=\"\" WEBCIENT_VERSION \">"); sprintf(szTmp, "Configuration Settings:
<PRE>" "Txn Monitor =" "%s
" "Database protocol =" %s
" "Max Connections =" %d
" "# of Delivery Threads =" %d
" "Max Pending Deliveries = %d
" , szTxnMonNames[Reg.eTxnMon], szDBNames[Reg.eDB_Protocol], Reg.dwMaxConnections, dwNumDeliveryThreads, dwDelBuffSize); strcat(szBuffer, szTmp); if (Reg.eTxnMon == COM) { sprintf(szTmp, "COM Single Pool =" "%s
", Reg.bCOM_SinglePool ? "YES" : "NO"); strcat(szBuffer, szTmp); } strcat(szBuffer, "</PRE>"); 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:
" " " face=\"Courier New\" color=\"blue\"><PRE>" "DB Server =" <INPUT NAME=\"db_server\" SIZE=20 VALUE=\"%s\">
" "DB User ID = <INPUT NAME=\"db_user\" SIZE=20 VALUE=\"%s\">
" "DB Password = <INPUT NAME=\"db_passwd\" SIZE=20 VALUE=\"%s\">
" "DB Name =" <INPUT NAME=\"db_name\" SIZE=20 VALUE=\"%s\">
" " "</PRE>" , 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:
" " " face=\"Courier New\" color=\"blue\"><PRE>" "DB Server =" %s
" </pre>
--	--

<pre> ID = %s
" "DB User Password = %s
" "DB "DB Name "</PRE>" , Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, Reg.szDbName); strcat(szBuffer, szTmp); sprintf(szTmp, "Please enter your Warehouse and District for this session:
" "<PRE>"); strcat(szBuffer, szTmp); strcat(szBuffer, "Warehouse ID = <INPUT NAME=\"w_id\" SIZE=4>
" "District ID = <INPUT NAME=\"d_id\" SIZE=2>
" "<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 </pre>	<pre> GetKeyValue(&ptr, "db_name", szDatabase, sizeof(szDatabase), NO_ERR); } // parse warehouse ID int w_id = GetIntKeyValue(&ptr, "w_id", ERR_HTML_ILL_FORMED, ERR_W_ID_INVALID); if (w_id < 1) throw new CWEBCLNT_ERR(ERR_W_ID_INVALID); // parse district ID int d_id = GetIntKeyValue(&ptr, "d_id", ERR_HTML_ILL_FORMED, ERR_D_ID_INVALID); if (d_id < 1 d_id > 10) throw new CWEBCLNT_ERR(ERR_D_ID_INVALID); iNewTerm = TermAdd(); Term.pClientData[iNewTerm].w_id = w_id; Term.pClientData[iNewTerm].d_id = d_id; try { if (Reg.eTxnMon == TUXEDO) Term.pClientData[iNewTerm].pTxn = pCTPCC_TUXEDO_new(); else if (Reg.eTxnMon == ENCINA) Term.pClientData[iNewTerm].pTxn = pCTPCC_ENCINA_new(); else if (Reg.eTxnMon == COM) Term.pClientData[iNewTerm].pTxn = pCTPCC_COM_new(Reg.bCOM_SinglePool); else if (Reg.eDB_Protocol == ODBC) Term.pClientData[iNewTerm].pTxn = pCTPCC_ODBC_new(szServer, szUser, szPassword, szMyComputerName, szDatabase); else if (Reg.eDB_Protocol == DBLIB) Term.pClientData[iNewTerm].pTxn = pCTPCC_DBLIB_new(szServer, szUser, szPassword, szMyComputerName, szDatabase); } catch (...) { TermDelete(iNewTerm); throw; // pass exception upward } MakeMainMenuForm(iNewTerm, Term.pClientData[iNewTerm].iSyncId, szBuffer); } /* FUNCTION: StatsCmd * * PURPOSE: This function returns to the browser the total number of active terminal ids. * This routine is for development/debugging purposes. * */ void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer) { int i; int iTotal; </pre>
--	---

```

EnterCriticalSection(&TermCriticalSection);

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

LeaveCriticalSection(&TermCriticalSection);

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

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

```

```

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

```

```

        {
            ERR_VERSION_MISMATCH,
            "Invalid version field. RTE and Web Client are probably
out of sync." },
        {
            ERR_W_ID_INVALID,
            "Invalid Warehouse ID."
        },
        {
            0,
            ""
        }
    };

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

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

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

/* FUNCTION: GetKeyValue
*
* PURPOSE:      This function parses a http formatted string for specific
key values.
*
* ARGUMENTS:   char                *pQueryString
http string from client browser
*
* pKey         char                key value to look for
*
* pValue       char                character array into which to place key's value
*
* iMax         int                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=\"TERMINID\"
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=\"SYNCID\"
VALUE=\"%d\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"

```

```

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

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

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

    c = wsprintf(szForm,
        "<HTML><HEAD><TITLE>TPC-C Stock
Level</TITLE></HEAD><FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
        "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">"
        "<INPUT TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"0\">"
        "<INPUT TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"TERMINID\"
VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\"
VALUE=\"%d\">"
        "<PRE><font face=\"Courier\">
Stock-Level<BR>"
        "Warehouse: %4.4d District: %2.2d<BR> <BR>",
        STOCK_LEVEL_FORM, iTermId,
Term.pClientData[iTermId].iSyncId,
Term.pClientData[iTermId].w_id,
Term.pClientData[iTermId].d_id);

    if ( bInput )
    {
        strcpy(szForm+c,
            "Stock Level Threshold: <INPUT
NAME=\"TT*\" SIZE=2><BR> <BR>"
            "low stock: </font><BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR>
" <BR> <BR> <BR> <BR> <BR> <BR> <BR>
" <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></PRE><HR>"
        VALUE="..NewOrder..">"
        VALUE="..Payment..">"
        VALUE="..Delivery..">"
        VALUE="..Order-Status..">"
        VALUE="..Stock-Level..">"
        VALUE="..Exit..">"

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

    if ( bInput )
    {
        c += sprintf(szForm+c, "Warehouse: %4.4d ",
Term.pClientData[iTermId].w_id);

        strcpy( szForm+c,
        "District: <INPUT NAME=\"DID*\"
SIZE=1>
Date:<BR>"
        "Customer: <INPUT NAME=\"CID*\"
SIZE=4> Name:
Credit: %Disc:<BR>"
        "Order Number:
Number of Lines:
W_tax: D_tax:<BR> <BR>"
        " Supp_W Item_Id Item Name Qty
Stock B/G Price Amount<BR>"
        " <INPUT NAME=\"SP00*\" SIZE=4>
<INPUT NAME=\"IID00*\" SIZE=6>
NAME=\"Qty00*\" SIZE=1><BR>"
        " <INPUT NAME=\"SP01*\" SIZE=4>
<INPUT NAME=\"IID01*\" SIZE=6>
NAME=\"Qty01*\" SIZE=1><BR>"
        " <INPUT NAME=\"SP02*\" SIZE=4>
<INPUT NAME=\"IID02*\" SIZE=6>
NAME=\"Qty02*\" SIZE=1><BR>"
        " <INPUT NAME=\"SP03*\" SIZE=4>
<INPUT NAME=\"IID03*\" SIZE=6>
NAME=\"Qty03*\" SIZE=1><BR>"
        " <INPUT NAME=\"SP04*\" SIZE=4>
<INPUT NAME=\"IID04*\" SIZE=6>
NAME=\"Qty04*\" SIZE=1><BR>"
        " <INPUT NAME=\"SP05*\" SIZE=4>
<INPUT NAME=\"IID05*\" SIZE=6>
NAME=\"Qty05*\" SIZE=1><BR>"
        " <INPUT NAME=\"SP06*\" SIZE=4>
<INPUT NAME=\"IID06*\" SIZE=6>
NAME=\"Qty06*\" SIZE=1><BR>"
        " <INPUT NAME=\"SP07*\" SIZE=4>
<INPUT NAME=\"IID07*\" SIZE=6>
NAME=\"Qty07*\" SIZE=1><BR>"
        " <INPUT NAME=\"SP08*\" SIZE=4>
<INPUT NAME=\"IID08*\" SIZE=6>
NAME=\"Qty08*\" SIZE=1><BR>"
        " <INPUT NAME=\"SP09*\" SIZE=4>
<INPUT NAME=\"IID09*\" SIZE=6>
NAME=\"Qty09*\" SIZE=1><BR>"
        " <INPUT NAME=\"SP10*\" SIZE=4>
<INPUT NAME=\"IID10*\" SIZE=6>
NAME=\"Qty10*\" SIZE=1><BR>"
        " <INPUT NAME=\"SP11*\" SIZE=4>
<INPUT NAME=\"IID11*\" SIZE=6>
NAME=\"Qty11*\" SIZE=1><BR>"
        " <INPUT NAME=\"SP12*\" SIZE=4>
<INPUT NAME=\"IID12*\" SIZE=6>
NAME=\"Qty12*\" SIZE=1><BR>"
        " <INPUT NAME=\"SP13*\" SIZE=4>
<INPUT NAME=\"IID13*\" SIZE=6>
NAME=\"Qty13*\" SIZE=1><BR>"
        " <INPUT NAME=\"SP14*\" SIZE=4>
<INPUT NAME=\"IID14*\" SIZE=6>
NAME=\"Qty14*\" SIZE=1><BR>"
        "Execution Status:
Total:<BR>"
        "</font></PRE><HR>"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\""
        "<INPUT TYPE=\"submit\" NAME=\"CMD\""
        "</FORM></HTML>"
        );
    }
    else
    {
        c += sprintf(szForm+c, "Warehouse: %4.4d 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,

"%5.2f      <BR>"

        "Order
Number: %8.8d Number of Lines: %2.2d W_tax: %5.2f D_tax: %5.2f
<BR> <BR>"

        " Supp_W
Item_Id Item Name Qty Stock B/G Price Amount<BR>",

100.0*pNewOrderData->c_discount,

        pNewOrderData->o_id,
pNewOrderData->o_ol_cnt,
100.0 * pNewOrderData->w_tax,
100.0 * pNewOrderData->d_tax);

        for(i=0; i<pNewOrderData->o_ol_cnt; i++)
        {
            c += sprintf(szForm+c, " %4.4d
%6.6d %%-24s %2.2d %3.3d %1.1s $%6.2f $%7.2f <BR>",

pNewOrderData->OL[i].ol_supply_w_id,

pNewOrderData->OL[i].ol_i_id,

pNewOrderData->OL[i].ol_i_name,

pNewOrderData->OL[i].ol_quantity,

pNewOrderData->OL[i].ol_stock,

pNewOrderData->OL[i].ol_brand_generic,

pNewOrderData->OL[i].ol_i_price,

pNewOrderData->OL[i].ol_amount );
        }
    }
    else
    {
        c += 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 committed. Total: $%8.2f ",
        pNewOrderData->total_amount);

    else
    c += sprintf(szForm+c, "Execution Status:
Item number is not valid. Total:");

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

        "</FORM></HTML>"
    );

    }

}

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

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

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

```

```

        , PAYMENT_FORM, iTermId,
Term.pClientData[iTermId].iSyncId);

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

    if ( bInput )
    {
        c += sprintf(szForm+c,
                    "<BR> <BR>Warehouse: %4.4d"
                    " District: <INPUT
NAME=\"DID*\" SIZE=1><BR> <BR> <BR> <BR> <BR>"
                    "Customer: <INPUT NAME=\"CID*\"
SIZE=4>"
                    "Cust-Warehouse: <INPUT NAME=\"CW1*\"
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: %4.4d
District: %2.2d<BR>"
                    "%-20s %-20s<BR>"
                    "%-20s %-20s<BR>"
                    "%5.5s-%4.4s<BR> <BR>"
                    "Customer: %4.4d Cust-Warehouse: %4.4d
Cust-District: %2.2d<BR>"
                    "Name: %-16s %-2s %-16s Since:
%-2.2d-%2.2d-%4.4d<BR>"
                    " %-20s Credit: %-2s<BR>"
                    , Term.pClientData[iTermId].w_id,
                    pPaymentData->w_street_1,
                    pPaymentData->w_street_2,
                    pPaymentData->w_street_2,
                    pPaymentData->w_city,
                    pPaymentData->w_state, pPaymentData->w_zip, pPaymentData->w_zip+5,
                    pPaymentData->d_city,
                    pPaymentData->d_state, pPaymentData->d_zip, pPaymentData->d_zip+5
                    , pPaymentData->c_id,
                    pPaymentData->c_d_id,
                    pPaymentData->c_first,
                    pPaymentData->c_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 += sprintf(szForm+c,
                    " %-20s %5.5s-%4.4s Phone:
%6.6s-%3.3s-%3.3s-%4.4s<BR> <BR>",
                    pPaymentData->c_city,
                    pPaymentData->c_state, pPaymentData->c_zip, pPaymentData->c_zip+5,
                    pPaymentData->c_phone,
                    pPaymentData->c_phone+9,
                    pPaymentData->c_phone+12 );
        c += sprintf(szForm+c,
                    "Amount Paid: $%7.2f New
Cust-Balance: $%14.2f<BR>"
                    "Credit Limit: $%13.2f<BR> <BR>"
                    , pPaymentData->h_amount,
                    pPaymentData->c_balance,
                    pPaymentData->c_credit_lim
                    );
        if ( pPaymentData->c_credit[0] == 'B' &&
pPaymentData->c_credit[1] == 'C' )
        c += sprintf(szForm+c,
                    "Cust-Data:
%-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: %4.4d ",
        ORDER_STATUS_FORM, iTermId,
        Term.pClientData[iTermId].iSyncId, Term.pClientData[iTermId].w_id);

```

```

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

```

```

        c += wsprintf(szForm+c,
            "District: %2.2d<BR>"
            "Customer: %4.4d Name: %-16s %-2s
%-16s<BR>",
            pOrderStatusData->d_id,
            pOrderStatusData->c_id,
            pOrderStatusData->c_first,
            pOrderStatusData->c_middle, pOrderStatusData->c_last);
        c += sprintf(szForm+c, "Cust-Balance: $%9.2f<BR>
<BR>",
            pOrderStatusData->c_balance);
        c += wsprintf(szForm+c,

```

```

        "Order-Number: %8.8d Entry-Date:
%2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d Carrier-Number: %2.2d<BR>"
        "Supply-W Item-Id Qty Amount
Delivery-Date<BR>",
        pOrderStatusData->o_id,
        pOrderStatusData->o_entry_d.day,
        pOrderStatusData->o_entry_d.month,
        pOrderStatusData->o_entry_d.year,
        pOrderStatusData->o_entry_d.hour,
        pOrderStatusData->o_entry_d.minute,
        pOrderStatusData->o_entry_d.second,
        pOrderStatusData->o_carrier_id);
        for(i=0; i< pOrderStatusData->o_ol_cnt; i++)
        {
            c += sprintf(szForm+c, " %4.4d %6.6d
%2.2d $%8.2f %2.2d-%2.2d-%4.4d<BR>",
            pOrderStatusData->OL[i].ol_supply_w_id,
            pOrderStatusData->OL[i].ol_i_id,
            pOrderStatusData->OL[i].ol_quantity,
            pOrderStatusData->OL[i].ol_amount,
            pOrderStatusData->OL[i].ol_delivery_d.day,
            pOrderStatusData->OL[i].ol_delivery_d.month,
            pOrderStatusData->OL[i].ol_delivery_d.year);
        }
        strncpy( szForm+c, szBR, (15-i)*5 );
        c += (15-i)*5;
        strcpy(szForm+c,
            "</font></PRE><HR><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
            "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"
            "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"
            "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Order-Status..\">"
            "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Stock-Level..\">"
            "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Exit..\">"
            "</BODY></FORM></HTML>");
    }
}

```

```

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

```

```

void MakeDeliveryForm(int iTermId, DELIVERY_DATA *pDeliveryData,
BOOL bInput, char *szForm)
{
    int            c;
    c = wsprintf(szForm,
        "<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD><BODY>"

```

```

        "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
        "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"0\">"
        "<INPUT TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"TERMINID\"
VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"SYNCDID\"
VALUE=\"%d\">"
        "<PRE><font face=\"Courier\">
Delivery<BR>"
        "Warehouse: %4.4d<BR> <BR>",
        (!bInput && (pDeliveryData->exec_status_code !=
eOK)) ? ERR_TYPE_DELIVERY_POST : 0,
        DELIVERY_FORM, iTermId,
Term.pClientData[iTermId].iSyncId, Term.pClientData[iTermId].w_id);

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

/* FUNCTION: ProcessNewOrderForm
*
* PURPOSE:      This function gets and validates the input data from the
new order form

```

```

*
*      filling in the required input variables. it then
calls the SQLNewOrder
*
*      transaction, constructs the output form and
writes it back to client
*
*      browser.
*/

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

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

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

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

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

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

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

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

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

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

/* FUNCTION: ProcessOrderStatusForm
*
* PURPOSE:      This function gets and validates the input data from the
Order Status

```

```

*                               form filling in the required input variables. It
then calls the
*                               SQLOrderStatus transaction, constructs the
output form and writes it
*                               back to client browser.
*
* ARGUMENTS:  EXTENSION_CONTROL_BLOCK      *pECB
passed in structure pointer from inetsrv.
*                               int
*                               iTermId  client browser terminal id
*/

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

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

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

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

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

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

    PDELIVERY_DATA pDelivery;

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

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

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

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

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

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

    PSTOCK_LEVEL_DATA    pStockLevel;

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

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

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

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

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

/* FUNCTION: GetNewOrderData
*

```

```

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

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

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

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

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

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

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

```

```

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

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

        pNewOrderData->o_ol_cnt = items;
    }

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

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

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

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

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

```

```

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

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

            _strupr( szTmp );
            if ( strlen(pPaymentData->c_last) > LAST_NAME_LEN
)
                throw new CWEBCLNT_ERR(
ERR_PAYMENT_LAST_NAME_TO_LONG );
            strcpy(pPaymentData->c_last, szTmp);
        }
        else
        {
            // parse customer id and verify that last name was NOT
entered
            GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_CLT_KEY);
            if ( szTmp[0] != 0 )
                throw new CWEBCLNT_ERR(
ERR_PAYMENT_CID_AND_CLT );
        }

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

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

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

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

    _strupr( szTmp );

```

```

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

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

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

/* FUNCTION: BOOL IsDecimal(char *ptr)
*
* PURPOSE:      This function determines if a string is a non-negative
decimal value.
*
*               It fails if any characters other than a series of numbers followed by
*
*               a decimal point, another series of numbers,
and a null terminator are present.
*
* ARGUMENTS:    char                *ptr        pointer to
string to check.
*
* RETURNS:      BOOL    FALSE    if string is not a valid
non-negative decimal value
*
*               TRUE    if
string is OK
*/
BOOL IsDecimal(char *ptr)
{
    char *dotptr;
    BOOL bValid;

```

```

if ( *ptr == 0 )
    return FALSE;

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

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

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

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

```

tpcc.def

LIBRARY TPCC.DLL

EXPORTS

```

GetExtensionVersion @1
HttpExtensionProc   @2
TerminateExtension  @3

```

tpcc.h

```

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

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

```

```

#define TP_MAX_RETRIES
50

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

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

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

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

    CTPCC_BASE *pTxn;
} CLIENTDATA, *PCLIENTDATA;

//This structure is used to define the operational interface for terminal id support
typedef struct _TERM
{
    int iNumEntries;
    //total allocated terminal array entries
    int iFreeList;
    //next available terminal array element or -1 if none
    int iMasterSyncId;
    //synchronization id
    CLIENTDATA *pClientData;
    //pointer to allocated client data
} TERM;

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

enum WEBERROR
{
    NO_ERR,
    ERR_COMMAND_UNDEFINED,

```

```

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

class CWEBCLNT_ERR : public CBaseErr
{
public:
    CWEBCLNT_ERR(WEBCERROR Err)
    {
        m_Error = Err;
        m_szTextDetail = NULL;
    }
};

m_SystemErr = 0;
m_szErrorText = NULL;
};

CWEBCLNT_ERR(WEBCERROR 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;
};

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

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

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

//function prototypes

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call,
LPVOID lpReserved);
void WriteMessageToEventLog(LPTSTR lpszMsg);
void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int
*pCmd, int *pFormId, int *pTermId, int *pSyncId);
void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char
*szBuffer);
void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId);
void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId);
void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int iError, int
iErrorType, char *szMsg, int iTermId);
void GetKeyValue(char **pQueryString, char *pKey, char *pValue, int iMax,
WEBCERROR err);
int GetIntKeyValue(char **pQueryString, char *pKey, WEBCERROR
NoKeyErr, WEBCERROR NotIntErr);
void TermInit(void);
void TermDeleteAll(void);
int TermAdd(void);
void TermDelete(int id);
void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int
iErrorNum, int iTermId, int iSyncId, char *szErrorText, char *szBuffer );

```

```

void MakeMainMenuForm(int iTermId, int iSyncId, char *szForm);
void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA
*pStockLevelData, BOOL bInput, char *szForm);
void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA
*pNewOrderData, BOOL bInput, char *szForm);
void MakePaymentForm(int iTermId, PAYMENT_DATA *pPaymentData,
BOOL bInput, char *szForm);
void MakeOrderStatusForm(int iTermId, ORDER_STATUS_DATA
*pOrderStatusData, BOOL bInput, char *szForm);
void MakeDeliveryForm(int iTermId, DELIVERY_DATA *pDeliveryData,
BOOL bInput, char *szForm);
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA
*pNewOrderData);
void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA
*pPaymentData);
void GetOrderStatusData(LPSTR lpszQueryString, ORDER_STATUS_DATA
*pOrderStatusData);
BOOL PostDeliveryInfo(short w_id, short o_carrier_id);
BOOL IsNumeric(char *ptr);
BOOL IsDecimal(char *ptr);
void DeliveryWorkerThread(void *ptr);

```

tpcc.rc

```

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

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

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

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

#ifdef _MAC
//
// Version
//

VS_VERSION_INFO VERSIONINFO
FILEVERSION 0,4,0,0
PRODUCTVERSION 0,4,0,0
FILEFLAGSMASK 0x3fL

```

```

#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x40004L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904b0"
        BEGIN
            VALUE "Comments", "TPC-C HTML DLL Server (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
////////////////////////////////////

#ifdef 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 definitions of structures specific to TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_com.h"

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

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

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

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

    m_bSinglePool = bSinglePool;

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

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

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

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

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

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

        hr = CoCreateInstance(CLSID_NewOrder, NULL,
CLSCTX_SERVER, IID_ITPCC, (void **)&m_pNewOrder);
        if (FAILED(hr))

```

```

        throw new CCOMERR(hr);

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

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

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

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

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

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

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

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

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

void CTPCC_COM::NewOrder()
{
    VARIANT vTxn_out;

    HRESULT hr = m_pNewOrder->NewOrder(m_vTxn, &vTxn_out);
    if (FAILED(hr))
        throw new CCOMERR( hr );
    memcpy(m_pTxn, (void
*)vTxn_out.parray->pvData, vTxn_out.parray->rgsabound[0].cElements);
    SafeArrayDestroy(vTxn_out.parray);

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

```

```

    }

void CTPCC_COM::Payment()
{
    VARIANT vTxn_out;

    HRESULT hr = m_pPayment->Payment(m_vTxn, &vTxn_out);
    if (FAILED(hr))
        throw new CCOMERR( hr );
    memcpy(m_pTxn, (void
*)vTxn_out.parray->pvData, vTxn_out.parray->rgsabound[0].cElements);
    SafeArrayDestroy(vTxn_out.parray);

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

void CTPCC_COM::StockLevel()
{
    VARIANT vTxn_out;

    HRESULT hr = m_pStockLevel->StockLevel(m_vTxn,
&vTxn_out);
    if (FAILED(hr))
        throw new CCOMERR( hr );
    memcpy(m_pTxn, (void
*)vTxn_out.parray->pvData, vTxn_out.parray->rgsabound[0].cElements);
    SafeArrayDestroy(vTxn_out.parray);

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

void CTPCC_COM::OrderStatus()
{
    VARIANT vTxn_out;

    HRESULT hr = m_pOrderStatus->OrderStatus(m_vTxn,
&vTxn_out);
    if (FAILED(hr))
        throw new CCOMERR( hr );
    memcpy(m_pTxn, (void
*)vTxn_out.parray->pvData, vTxn_out.parray->rgsabound[0].cElements);
    SafeArrayDestroy(vTxn_out.parray);

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

```

tpcc_com.h

```

/*      FILE:          TPCC_COM.H
*                               Microsoft TPC-C Kit Ver.
4.20.000
*                               Copyright Microsoft, 1999
*
*      All Rights Reserved
*
*                               not yet audited
*
*      PURPOSE:        Header file for TPC-C COM+ class
implementation.
*
*      Change history:
*                               4.20.000 - first version

```

```

*/

#pragma once

#include <stdio.h>
#include "..\..\tpcc_com_ps\src\tpcc_com_ps.h"

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

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

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

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

    int m_hr;
    int m_iErrorType;
    int m_iError;

    // A CCOMERR class can impersonate another class,
    // which happens if the error
    // was not actually a COM Services error, but was simply
    // transmitted back via COM.
    int ErrorType()
    {
        if (m_iErrorType == 0)
            return ERR_TYPE_COM;
        else
            return m_iErrorType;
    }

    int ErrorNum() {return m_hr;}

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

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

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

    struct COM_DATA
    {
        int ErrorType;
        int error;
        union
        {
            NEW_ORDER_DATA
            PAYMENT_DATA
            DELIVERY_DATA
            STOCK_LEVEL_DATA
            ORDER_STATUS_DATA
        } u;
    } *m_pTxn;

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

    inline PNEW_ORDER_DATA
    BuffAddr_NewOrder() { return &m_pTxn->u.NewOrder;
    };

    inline PPAYMENT_DATA
    BuffAddr_Payment() { return &m_pTxn->u.Payment; };

    inline PDELIVERY_DATA
    BuffAddr_Delivery() { return &m_pTxn->u.Delivery; };

    inline PSTOCK_LEVEL_DATA BuffAddr_StockLevel()
    { return &m_pTxn->u.StockLevel; };

    inline PORDER_STATUS_DATA
    BuffAddr_OrderStatus() { return &m_pTxn->u.OrderStatus; };

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

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

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

    typedef CTPCC_COM* (TYPE_CTPCC_COM)(BOOL);

```

tpcc_com_all.dsp

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

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

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

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

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

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

```
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib
odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
```

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

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

!ENDIF
```

```
# Begin Target

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

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

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

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

SOURCE=. \src\tpcc_com_all.idl

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

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

BuildCmds= \
```

```

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

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

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

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

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

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

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

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

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

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

!ENDIF

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

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

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

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

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

```

tpcc_com_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 5.03.0280 */
/* at Sat Apr 08 16:40:10 2000 */
/*
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADING( )

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

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

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

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

#ifndef tpcc_com_ps_h_
#define tpcc_com_ps_h_

/* Forward Declarations */

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

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

#ifndef __cplusplus
extern "C" {
#endif

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

/* interface __MIDL_itf_tpcc_com_ps_0000 */
/* [local] */

```

```

extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

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

EXTERN_C const IID IID_ITPCC;

#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 __RPC_FAR *txn_out) = 0;

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

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

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

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

    virtual HRESULT __stdcall CallSetComplete( void) = 0;

};

#else /* C style interface */

typedef struct ITPCCVtbl
{
    BEGIN_INTERFACE

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

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

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

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

```

```

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

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

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

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

        HRESULT ( __stdcall __RPC_FAR *CallSetComplete )(
            ITPCC __RPC_FAR * This);

    END_INTERFACE
} ITPCCVtbl;

interface ITPCC
{
    CONST_VTBL struct ITPCCVtbl __RPC_FAR *lpVtbl;
};

#ifndef COBJMACROS

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

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

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

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

#define ITPCC_Payment(This,txn_in,txn_out) \
    (This)->lpVtbl->Payment(This,txn_in,txn_out)

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

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

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

#define ITPCC_CallSetComplete(This) \
    (This)->lpVtbl->CallSetComplete(This)

#endif /* COBJMACROS */

#endif /* C style interface */

```

```
HRESULT __stdcall ITPCC_NewOrder_Proxy(
    ITPCC_RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);
```

```
void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);
```

```
HRESULT __stdcall ITPCC_Payment_Proxy(
    ITPCC_RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);
```

```
void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);
```

```
HRESULT __stdcall ITPCC_Delivery_Proxy(
    ITPCC_RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);
```

```
void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);
```

```
HRESULT __stdcall ITPCC_StockLevel_Proxy(
    ITPCC_RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);
```

```
void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);
```

```
HRESULT __stdcall ITPCC_OrderStatus_Proxy(
    ITPCC_RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);
```

```
void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);
```

```
HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC_RPC_FAR * This);
```

```
void __RPC_STUB ITPCC_CallSetComplete_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);
```

```
#endif /* __ITPCC_INTERFACE_DEFINED__ */
```

```
/* Additional Prototypes for ALL interfaces */
```

```
unsigned long __RPC_USER VARIANT_UserSize( unsigned long
    __RPC_FAR *, unsigned long , VARIANT __RPC_FAR * );
unsigned char __RPC_FAR * __RPC_USER VARIANT_UserMarshal(
    unsigned long __RPC_FAR *, unsigned char __RPC_FAR *, VARIANT
    __RPC_FAR * );
unsigned char __RPC_FAR * __RPC_USER
    VARIANT_UserUnmarshal(unsigned long __RPC_FAR *, unsigned char
    __RPC_FAR *, VARIANT __RPC_FAR * );
void __RPC_USER VARIANT_UserFree( unsigned long
    __RPC_FAR *, VARIANT __RPC_FAR * );
```

```
/* end of Additional Prototypes */
```

```
#ifdef __cplusplus
}
#endif
```

```
#endif
```

tpcc_com_ps.idl

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

```
// Forward declare all types defined
```

```
interface ITPCC;
import "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
        (
            [in]
            VARIANT txn_in,
            [out]
            VARIANT *txn_out
        );

}; // interface ITPCC

```

tpcc_com_ps_i.c

```

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

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */

/* File created by MIDL compiler version 5.03.0280 */
/* at Sat Apr 08 16:40:10 2000
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
    DECLSPEC_UUID(), MIDL_INTERFACE()
*/

```

```

//@@@MIDL_FILE_HEADING( )

#ifdef _M_IA64 && !defined(_M_AXP64)

#ifndef __cplusplus
extern "C" {
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,
0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif // !defined(_M_IA64) && !defined(_M_AXP64)

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

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

```

```

/* link this file in with the server and any clients */

/* File created by MIDL compiler version 5.03.0280 */
/* at Sat Apr 08 16:40:10 2000
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext,
robust
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if defined(_M_IA64) || defined(_M_AXP64)

#ifdef __cplusplus
extern "C" {
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,
0xE0,0x8B);

```

```

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif /* defined(_M_IA64) || defined(_M_AXP64) */

tpcc_com_ps_p.c

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

/* this ALWAYS GENERATED file contains the proxy stub code */

/* File created by MIDL compiler version 5.03.0280 */
/* at Sat Apr 08 16:40:10 2000
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if !defined(_M_IA64) && !defined(_M_AXP64)
#define USE_STUBLESS_PROXY

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

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

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 997
#define PROC_FORMAT_STRING_SIZE 193
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 1

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

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

extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;

```

```

extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,

GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,

GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,

GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

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

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

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

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

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

extern const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

static const MIDL_STUB_DESC Object_StubDesc =
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.3.280 */
    0,
    UserMarshalRoutines,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* Reserved3 */
    0, /* Reserved4 */
    0 /* Reserved5 */
};

#pragma data_seg(".rdata")

static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
{
    {
        VARIANT_UserSize,
        VARIANT_UserMarshal,
        VARIANT_UserUnmarshal,
        VARIANT_UserFree
    }
};

#ifdef __RPC_WIN32__
#error Invalid build platform for this stub.
#endif

#ifdef !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.
#endif

```

```

#error This app will die there with the RPC_X_WRONG_STUB_VERSION
error.
#endif

static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
{
    0,
    {

        /* Procedure NewOrder */

        FC_AUTO_HANDLE */
        0x33,
        /*
        0x6c,
        /* Old Flags: object,
        Oi2 */
        /* 2 */ NdrFcLong( 0x0 ), /* 0 */
        /* 6 */ NdrFcShort( 0x3 ), /* 3 */
        #ifndef _ALPHA_
        #ifndef _PPC_
        #if !defined(_MIPS_)
        /* 8 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
        #else
        NdrFcShort( 0x20 ), /* MIPS Stack
        size/offset = 32 */
        #endif
        #else
        NdrFcShort( 0x20 ), /* PPC Stack size/offset
        = 32 */
        #endif
        #else
        NdrFcShort( 0x28 ), /* Alpha Stack
        size/offset = 40 */
        #endif
        /* 10 */ NdrFcShort( 0x0 ), /* 0 */
        /* 12 */ NdrFcShort( 0x8 ), /* 8 */
        /* 14 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has
        return, */
        0x3, /* 3 */

        /* Parameter txn_in */

        /* 16 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
        #ifndef _ALPHA_
        #ifndef _PPC_
        #if !defined(_MIPS_)
        /* 18 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
        #else
        NdrFcShort( 0x8 ), /* MIPS Stack
        size/offset = 8 */
        #endif
        #else
        NdrFcShort( 0x8 ), /* PPC Stack size/offset
        = 8 */
        #endif
        #else
        NdrFcShort( 0x8 ), /* Alpha Stack
        size/offset = 8 */
        #endif
        /* 20 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

        /* Parameter txn_out */

        /* 22 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out,
        simple ref, srv alloc size=16 */
        #ifndef _ALPHA_
        #ifndef _PPC_
        #if !defined(_MIPS_)
        /* 24 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
        #else
        NdrFcShort( 0x18 ), /* MIPS Stack
        size/offset = 24 */
        #endif
        #else
        NdrFcShort( 0x18 ), /* PPC Stack size/offset
        = 24 */
        #endif
        #else
        NdrFcShort( 0x18 ), /* Alpha Stack
        size/offset = 24 */
        #endif
        /* 26 */ NdrFcShort( 0x3da ), /* Type Offset=986 */

        /* Return value */

        /* 28 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
        #ifndef _ALPHA_
        #ifndef _PPC_
        #if !defined(_MIPS_)
        /* 30 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
        #else
        NdrFcShort( 0x1c ), /* MIPS Stack
        size/offset = 28 */
        #endif
        #else
        NdrFcShort( 0x1c ), /* PPC Stack size/offset
        = 28 */
        #endif
        #else
        NdrFcShort( 0x20 ), /* Alpha Stack
        size/offset = 32 */
        #endif
        /* 32 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

        /* Procedure Payment */

        /* 34 */ 0x33, /* FC_AUTO_HANDLE */
        0x6c, /* Old Flags: object,
        Oi2 */
        /* 36 */ NdrFcLong( 0x0 ), /* 0 */
        /* 40 */ NdrFcShort( 0x4 ), /* 4 */
        #ifndef _ALPHA_
        #ifndef _PPC_
        #if !defined(_MIPS_)
        /* 42 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
        #else
        NdrFcShort( 0x20 ), /* MIPS Stack
        size/offset = 32 */
        #endif
        #else
        NdrFcShort( 0x20 ), /* PPC Stack size/offset
        = 32 */
        #endif
        #else
        NdrFcShort( 0x28 ), /* Alpha Stack
        size/offset = 40 */
        #endif
        /* 44 */ NdrFcShort( 0x0 ), /* 0 */
        /* 46 */ NdrFcShort( 0x8 ), /* 8 */
        /* 48 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has
        return, */
        0x3, /* 3 */

        /* Parameter txn_in */

```

```

/* 50 */  NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 52 */  NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
NdrFcShort( 0x8 ), /* MIPS Stack

size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* PPC Stack size/offset

= 8 */
#endif
#else
NdrFcShort( 0x8 ), /* Alpha Stack

size/offset = 8 */
#endif
/* 54 */  NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Parameter txn_out */

/* 56 */  NdrFcShort( 0x4113 ), /* Flags: must size, must free, out,
simple ref, srv alloc size=16 */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 58 */  NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
NdrFcShort( 0x18 ), /* MIPS Stack

size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* PPC Stack size/offset

= 24 */
#endif
#else
NdrFcShort( 0x18 ), /* Alpha Stack

size/offset = 24 */
#endif
/* 60 */  NdrFcShort( 0x3da ), /* Type Offset=986 */

/* Return value */

/* 62 */  NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 64 */  NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
NdrFcShort( 0x1c ), /* MIPS Stack

size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /* PPC Stack size/offset

= 28 */
#endif
#else
NdrFcShort( 0x20 ), /* Alpha Stack

size/offset = 32 */
#endif
/* 66 */  0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure Delivery */

/* 68 */  0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */

```

```

/* 70 */  NdrFcLong( 0x0 ), /* 0 */
/* 74 */  NdrFcShort( 0x5 ), /* 5 */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 76 */  NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
NdrFcShort( 0x20 ), /* MIPS Stack

size/offset = 32 */
#endif
#else
NdrFcShort( 0x20 ), /* PPC Stack size/offset

= 32 */
#endif
#else
NdrFcShort( 0x20 ), /* Alpha Stack

size/offset = 40 */
#endif
/* 78 */  NdrFcShort( 0x0 ), /* 0 */
/* 80 */  NdrFcShort( 0x8 ), /* 8 */
/* 82 */  0x7, /* Oi2 Flags: srv must size, clt must size, has
return, */
0x3, /* 3 */

/* Parameter txn_in */

/* 84 */  NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 86 */  NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
NdrFcShort( 0x8 ), /* MIPS Stack

size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* PPC Stack size/offset

= 8 */
#endif
#else
NdrFcShort( 0x8 ), /* Alpha Stack

size/offset = 8 */
#endif
/* 88 */  NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Parameter txn_out */

/* 90 */  NdrFcShort( 0x4113 ), /* Flags: must size, must free, out,
simple ref, srv alloc size=16 */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 92 */  NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
NdrFcShort( 0x18 ), /* MIPS Stack

size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* PPC Stack size/offset

= 24 */
#endif
#else
NdrFcShort( 0x18 ), /* Alpha Stack

size/offset = 24 */
#endif
/* 94 */  NdrFcShort( 0x3da ), /* Type Offset=986 */

/* Return value */

```

```

/* 96 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 98 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
NdrFcShort( 0x1c ), /* MIPS Stack
size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /* PPC Stack size/offset
= 28 */
#endif
#else
NdrFcShort( 0x20 ), /* Alpha Stack
size/offset = 32 */
#endif
/* 100 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure StockLevel */

/* 102 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */
/* 104 */ NdrFcLong( 0x0 ), /* 0 */
/* 108 */ NdrFcShort( 0x6 ), /* 6 */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 110 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
NdrFcShort( 0x20 ), /* MIPS Stack
size/offset = 32 */
#endif
#else
NdrFcShort( 0x20 ), /* PPC Stack size/offset
= 32 */
#endif
#else
NdrFcShort( 0x28 ), /* Alpha Stack
size/offset = 40 */
#endif
/* 112 */ NdrFcShort( 0x0 ), /* 0 */
/* 114 */ NdrFcShort( 0x8 ), /* 8 */
/* 116 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has
return, */
0x3, /* 3 */

/* Parameter txn_in */

/* 118 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 120 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
NdrFcShort( 0x8 ), /* MIPS Stack
size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* PPC Stack size/offset
= 8 */
#endif
#else
NdrFcShort( 0x8 ), /* Alpha Stack
size/offset = 8 */
#endif
#endif
#endif
/* 122 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Parameter txn_out */

/* 124 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out,
simple ref, srv alloc size=16 */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 126 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
NdrFcShort( 0x18 ), /* MIPS Stack
size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* PPC Stack size/offset
= 24 */
#endif
#else
NdrFcShort( 0x18 ), /* Alpha Stack
size/offset = 24 */
#endif
#endif
/* 128 */ NdrFcShort( 0x3da ), /* Type Offset=986 */

/* Return value */

/* 130 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 132 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
NdrFcShort( 0x1c ), /* MIPS Stack
size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /* PPC Stack size/offset
= 28 */
#endif
#else
NdrFcShort( 0x20 ), /* Alpha Stack
size/offset = 32 */
#endif
#endif
/* 134 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure OrderStatus */

/* 136 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */
/* 138 */ NdrFcLong( 0x0 ), /* 0 */
/* 142 */ NdrFcShort( 0x7 ), /* 7 */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 144 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
NdrFcShort( 0x20 ), /* MIPS Stack
size/offset = 32 */
#endif
#else
NdrFcShort( 0x20 ), /* PPC Stack size/offset
= 32 */
#endif
#else
NdrFcShort( 0x20 ), /* Alpha Stack
size/offset = 32 */
#endif
#endif

```

```

NdrFcShort( 0x28 ), /* Alpha Stack
size/offset = 40 */
#endif
/* 146 */ NdrFcShort( 0x0 ), /* 0 */
/* 148 */ NdrFcShort( 0x8 ), /* 8 */
/* 150 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has
return, */
0x3, /* 3 */

/* Parameter txn_in */

/* 152 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 154 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
NdrFcShort( 0x8 ), /* MIPS Stack
size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* PPC Stack size/offset
= 8 */
#endif
#else
NdrFcShort( 0x8 ), /* Alpha Stack
size/offset = 8 */
#endif
/* 156 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Parameter txn_out */

/* 158 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out,
simple ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 160 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
NdrFcShort( 0x18 ), /* MIPS Stack
size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* PPC Stack size/offset
= 24 */
#endif
#else
NdrFcShort( 0x18 ), /* Alpha Stack
size/offset = 24 */
#endif
/* 162 */ NdrFcShort( 0x3da ), /* Type Offset=986 */

/* Return value */

/* 164 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 166 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
NdrFcShort( 0x1c ), /* MIPS Stack
size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /* PPC Stack size/offset
= 28 */
#endif
#else
#endif

```

```

NdrFcShort( 0x20 ), /* Alpha Stack
size/offset = 32 */
#endif
/* 168 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure CallSetComplete */

/* 170 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */
/* 172 */ NdrFcLong( 0x0 ), /* 0 */
/* 176 */ NdrFcShort( 0x8 ), /* 8 */
#ifndef _ALPHA_
/* 178 */ NdrFcShort( 0x8 ), /* x86, MIPS, PPC Stack size/offset = 8 */
#else
NdrFcShort( 0x10 ), /* Alpha Stack
size/offset = 16 */
#endif
/* 180 */ NdrFcShort( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x8 ), /* 8 */
/* 184 */ 0x4, /* Oi2 Flags: has return, */
0x1, /* 1 */

/* Return value */

/* 186 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
/* 188 */ NdrFcShort( 0x4 ), /* x86, MIPS, PPC Stack size/offset = 4 */
#else
NdrFcShort( 0x8 ), /* Alpha Stack
size/offset = 8 */
#endif
/* 190 */ 0x8, /* FC_LONG */
0x0, /* 0 */

0x0

}
};

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
0,
{
NdrFcShort( 0x0 ), /* 0 */
/* 2 */
0x12, 0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0x3b0 ), /* Offset= 944 (948) */
/* 6 */
0x2b, /*
FC_NON_ENCAPSULATED_UNION */
0x9, /* FC_ULONG */
/* 8 */ 0x7, /* Corr desc: FC_USHORT */
0x0, /* */
/* 10 */ NdrFcShort( 0xffff ), /* -8 */
/* 12 */ NdrFcShort( 0x2 ), /* Offset= 2 (14) */
/* 14 */ NdrFcShort( 0x10 ), /* 16 */
/* 16 */ NdrFcShort( 0x2b ), /* 43 */
/* 18 */ NdrFcLong( 0x3 ), /* 3 */
/* 22 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 24 */ NdrFcLong( 0x11 ), /* 17 */
/* 28 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE */
/* 30 */ NdrFcLong( 0x2 ), /* 2 */
/* 34 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 36 */ NdrFcLong( 0x4 ), /* 4 */
/* 40 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT */
/* 42 */ NdrFcLong( 0x5 ), /* 5 */

```

```

/* 46 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 48 */ NdrFcLong( 0xb ), /* 11 */
/* 52 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 54 */ NdrFcLong( 0xa ), /* 10 */
/* 58 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 60 */ NdrFcLong( 0x6 ), /* 6 */
/* 64 */ NdrFcShort( 0xd6 ), /* Offset= 214 (278) */
/* 66 */ NdrFcLong( 0x7 ), /* 7 */
/* 70 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 72 */ NdrFcLong( 0x8 ), /* 8 */
/* 76 */ NdrFcShort( 0xd0 ), /* Offset= 208 (284) */
/* 78 */ NdrFcLong( 0xd ), /* 13 */
/* 82 */ NdrFcShort( 0xe2 ), /* Offset= 226 (308) */
/* 84 */ NdrFcLong( 0x9 ), /* 9 */
/* 88 */ NdrFcShort( 0xee ), /* Offset= 238 (326) */
/* 90 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 94 */ NdrFcShort( 0xfa ), /* Offset= 250 (344) */
/* 96 */ NdrFcLong( 0x24 ), /* 36 */
/* 100 */ NdrFcShort( 0x308 ), /* Offset= 776 (876) */
/* 102 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 106 */ NdrFcShort( 0x302 ), /* Offset= 770 (876) */
/* 108 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 112 */ NdrFcShort( 0x300 ), /* Offset= 768 (880) */
/* 114 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 118 */ NdrFcShort( 0x2fe ), /* Offset= 766 (884) */
/* 120 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 124 */ NdrFcShort( 0x2fc ), /* Offset= 764 (888) */
/* 126 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 130 */ NdrFcShort( 0x2fa ), /* Offset= 762 (892) */
/* 132 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 136 */ NdrFcShort( 0x2f8 ), /* Offset= 760 (896) */
/* 138 */ NdrFcLong( 0x400b ), /* 16395 */
/* 142 */ NdrFcShort( 0x2e6 ), /* Offset= 742 (884) */
/* 144 */ NdrFcLong( 0x400a ), /* 16394 */
/* 148 */ NdrFcShort( 0x2e4 ), /* Offset= 740 (888) */
/* 150 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 154 */ NdrFcShort( 0x2ea ), /* Offset= 746 (900) */
/* 156 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 160 */ NdrFcShort( 0x2e0 ), /* Offset= 736 (896) */
/* 162 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 166 */ NdrFcShort( 0x2e2 ), /* Offset= 738 (904) */
/* 168 */ NdrFcLong( 0x400d ), /* 16397 */
/* 172 */ NdrFcShort( 0x2e0 ), /* Offset= 736 (908) */
/* 174 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 178 */ NdrFcShort( 0x2de ), /* Offset= 734 (912) */
/* 180 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 184 */ NdrFcShort( 0x2dc ), /* Offset= 732 (916) */
/* 186 */ NdrFcLong( 0x400c ), /* 16396 */
/* 190 */ NdrFcShort( 0x2da ), /* Offset= 730 (920) */
/* 192 */ NdrFcLong( 0x10 ), /* 16 */
/* 196 */ NdrFcShort( 0x8002 ), /* Simple arm type: FC_CHAR */
/* 198 */ NdrFcLong( 0x12 ), /* 18 */
/* 202 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 204 */ NdrFcLong( 0x13 ), /* 19 */
/* 208 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 210 */ NdrFcLong( 0x16 ), /* 22 */
/* 214 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 216 */ NdrFcLong( 0x17 ), /* 23 */
/* 220 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 222 */ NdrFcLong( 0xe ), /* 14 */
/* 226 */ NdrFcShort( 0x2be ), /* Offset= 702 (928) */
/* 228 */ NdrFcLong( 0x400e ), /* 16398 */
/* 232 */ NdrFcShort( 0x2c4 ), /* Offset= 708 (940) */
/* 234 */ NdrFcLong( 0x4010 ), /* 16400 */
/* 238 */ NdrFcShort( 0x2c2 ), /* Offset= 706 (944) */
/* 240 */ NdrFcLong( 0x4012 ), /* 16402 */

/* 244 */ NdrFcShort( 0x280 ), /* Offset= 640 (884) */
/* 246 */ NdrFcLong( 0x4013 ), /* 16403 */
/* 250 */ NdrFcShort( 0x27e ), /* Offset= 638 (888) */
/* 252 */ NdrFcLong( 0x4016 ), /* 16406 */
/* 256 */ NdrFcShort( 0x278 ), /* Offset= 632 (888) */
/* 258 */ NdrFcLong( 0x4017 ), /* 16407 */
/* 262 */ NdrFcShort( 0x272 ), /* Offset= 626 (888) */
/* 264 */ NdrFcLong( 0x0 ), /* 0 */
/* 268 */ NdrFcShort( 0x0 ), /* Offset= 0 (268) */
/* 270 */ NdrFcLong( 0x1 ), /* 1 */
/* 274 */ NdrFcShort( 0x0 ), /* Offset= 0 (274) */
/* 276 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (275) */
/* 278 */

0x15, /* FC_STRUCT */
0x7, /* 7 */

/* 280 */ NdrFcShort( 0x8 ), /* 8 */
/* 282 */ 0xb, /* FC_HYPER */
0x5b, /* FC_END */

/* 284 */

0x12, 0x0, /* FC_UP */
/* 286 */ NdrFcShort( 0xc ), /* Offset= 12 (298) */
/* 288 */

0x1b, /* FC_CARRAY */
0x1, /* 1 */

/* 290 */ NdrFcShort( 0x2 ), /* 2 */
/* 292 */ 0x9, /* Corr desc: FC_ULONG */
0x0, /* */

/* 294 */ NdrFcShort( 0xfffc ), /* -4 */
/* 296 */ 0x6, /* FC_SHORT */
0x5b, /* FC_END */

/* 298 */

0x17, /* FC_CSTRUCT */
0x3, /* 3 */

/* 300 */ NdrFcShort( 0x8 ), /* 8 */
/* 302 */ NdrFcShort( 0xffffffff2 ), /* Offset= -14 (288) */
/* 304 */ 0x8, /* FC_LONG */
0x8, /* FC_LONG */
/* 306 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */

/* 308 */

0x2f, /* FC_IP */
0x5a, /*

FC_CONSTANT_IID */
/* 310 */ NdrFcLong( 0x0 ), /* 0 */
/* 314 */ NdrFcShort( 0x0 ), /* 0 */
/* 316 */ NdrFcShort( 0x0 ), /* 0 */
/* 318 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 320 */ 0x0, /* 0 */
0x0, /* 0 */
/* 322 */ 0x0, /* 0 */
0x0, /* 0 */
/* 324 */ 0x0, /* 0 */
0x46, /* 70 */

0x2f, /* FC_IP */
0x5a, /*

FC_CONSTANT_IID */
/* 328 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 332 */ NdrFcShort( 0x0 ), /* 0 */
/* 334 */ NdrFcShort( 0x0 ), /* 0 */
/* 336 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 338 */ 0x0, /* 0 */
0x0, /* 0 */
/* 340 */ 0x0, /* 0 */
0x0, /* 0 */
/* 342 */ 0x0, /* 0 */

```

<pre> /* 344 */ 0x46, /* 70 */ /* 345 */ 0x12, 0x10, /* FC_UP */ [pointer_deref] */ /* 346 */ NdrFcShort(0x2), /* Offset= 2 (348) */ /* 348 */ 0x12, 0x0, /* FC_UP */ /* 350 */ NdrFcShort(0x1fc), /* Offset= 508 (858) */ /* 352 */ 0x2a, /* FC_ENCAPSULATED_UNION */ 0x49, /* 73 */ /* 354 */ NdrFcShort(0x18), /* 24 */ /* 356 */ NdrFcShort(0xa), /* 10 */ /* 358 */ NdrFcLong(0x8), /* 8 */ /* 362 */ NdrFcShort(0x58), /* Offset= 88 (450) */ /* 364 */ NdrFcLong(0xd), /* 13 */ /* 368 */ NdrFcShort(0x78), /* Offset= 120 (488) */ /* 370 */ NdrFcLong(0x9), /* 9 */ /* 374 */ NdrFcShort(0x94), /* Offset= 148 (522) */ /* 376 */ NdrFcLong(0xc), /* 12 */ /* 380 */ NdrFcShort(0xbc), /* Offset= 188 (568) */ /* 382 */ NdrFcLong(0x24), /* 36 */ /* 386 */ NdrFcShort(0x114), /* Offset= 276 (662) */ /* 388 */ NdrFcLong(0x800d), /* 32781 */ /* 392 */ NdrFcShort(0x130), /* Offset= 304 (696) */ /* 394 */ NdrFcLong(0x10), /* 16 */ /* 398 */ NdrFcShort(0x148), /* Offset= 328 (726) */ /* 400 */ NdrFcLong(0x2), /* 2 */ /* 404 */ NdrFcShort(0x160), /* Offset= 352 (756) */ /* 406 */ NdrFcLong(0x3), /* 3 */ /* 410 */ NdrFcShort(0x178), /* Offset= 376 (786) */ /* 412 */ NdrFcLong(0x14), /* 20 */ /* 416 */ NdrFcShort(0x190), /* Offset= 400 (816) */ /* 418 */ NdrFcShort(0xffffffff), /* Offset= -1 (417) */ /* 420 */ 0x1b, /* FC_CARRAY */ 0x3, /* 3 */ /* 422 */ NdrFcShort(0x4), /* 4 */ /* 424 */ 0x19, /* Corr desc: field pointer, FC_ULONG */ 0x0, /* */ /* 426 */ NdrFcShort(0x0), /* 0 */ /* 428 */ 0x4b, /* FC_PP */ 0x5c, /* FC_PAD */ /* 430 */ 0x48, /* FC_VARIABLE_REPEAT */ 0x49, /* FC_FIXED_OFFSET */ /* 432 */ NdrFcShort(0x4), /* 4 */ /* 434 */ NdrFcShort(0x0), /* 0 */ /* 436 */ NdrFcShort(0x1), /* 1 */ /* 438 */ NdrFcShort(0x0), /* 0 */ /* 440 */ NdrFcShort(0x0), /* 0 */ /* 442 */ 0x12, 0x0, /* FC_UP */ /* 444 */ NdrFcShort(0xffffffff6e), /* Offset= -146 (298) */ /* 446 */ 0x5b, /* FC_END */ 0x8, /* FC_LONG */ /* 448 */ 0x5c, /* FC_PAD */ 0x5b, /* FC_END */ /* 450 */ 0x16, /* FC_PSTRUCT */ 0x3, /* 3 */ /* 452 */ NdrFcShort(0x8), /* 8 */ /* 454 */ </pre>	<pre> 0x4b, /* FC_PP */ 0x5c, /* FC_PAD */ /* 456 */ 0x46, /* FC_NO_REPEAT */ 0x5c, /* FC_PAD */ /* 458 */ NdrFcShort(0x4), /* 4 */ /* 460 */ NdrFcShort(0x4), /* 4 */ /* 462 */ 0x11, 0x0, /* FC_RP */ /* 464 */ NdrFcShort(0xffffffffd4), /* Offset= -44 (420) */ /* 466 */ 0x5b, /* FC_END */ 0x8, /* FC_LONG */ /* 468 */ 0x8, /* FC_LONG */ 0x5b, /* FC_END */ /* 470 */ 0x21, /* FC_BOGUS_ARRAY */ 0x3, /* 3 */ /* 472 */ NdrFcShort(0x0), /* 0 */ /* 474 */ 0x19, /* Corr desc: field pointer, FC_ULONG */ 0x0, /* */ /* 476 */ NdrFcShort(0x0), /* 0 */ /* 478 */ NdrFcLong(0xffffffff), /* -1 */ /* 482 */ 0x4c, /* FC_EMBEDDED_COMPLEX */ 0x0, /* 0 */ /* 484 */ NdrFcShort(0xffffffff50), /* Offset= -176 (308) */ /* 486 */ 0x5c, /* FC_PAD */ 0x5b, /* FC_END */ /* 488 */ 0x1a, /* FC_BOGUS_STRUCT */ 0x3, /* 3 */ /* 490 */ NdrFcShort(0x8), /* 8 */ /* 492 */ NdrFcShort(0x0), /* 0 */ /* 494 */ NdrFcShort(0x6), /* Offset= 6 (500) */ /* 496 */ 0x8, /* FC_LONG */ 0x36, /* FC_POINTER */ /* 498 */ 0x5c, /* FC_PAD */ 0x5b, /* FC_END */ /* 500 */ 0x11, 0x0, /* FC_RP */ /* 502 */ NdrFcShort(0xffffffffe0), /* Offset= -32 (470) */ /* 504 */ 0x21, /* FC_BOGUS_ARRAY */ 0x3, /* 3 */ /* 506 */ NdrFcShort(0x0), /* 0 */ /* 508 */ 0x19, /* Corr desc: field pointer, FC_ULONG */ 0x0, /* */ /* 510 */ NdrFcShort(0x0), /* 0 */ /* 512 */ NdrFcLong(0xffffffff), /* -1 */ /* 516 */ 0x4c, /* FC_EMBEDDED_COMPLEX */ 0x0, /* 0 */ /* 518 */ NdrFcShort(0xffffffff40), /* Offset= -192 (326) */ /* 520 */ 0x5c, /* FC_PAD */ 0x5b, /* FC_END */ /* 522 */ 0x1a, /* FC_BOGUS_STRUCT */ 0x3, /* 3 */ /* 524 */ NdrFcShort(0x8), /* 8 */ /* 526 */ NdrFcShort(0x0), /* 0 */ /* 528 */ NdrFcShort(0x6), /* Offset= 6 (534) */ /* 530 */ 0x8, /* FC_LONG */ 0x36, /* FC_POINTER */ /* 532 */ 0x5c, /* FC_PAD */ 0x5b, /* FC_END */ </pre>
---	---

```

/* 534 */
                                0x11, 0x0, /* FC_RP */
/* 536 */ NdrFcShort( 0xfffffe0 ), /* Offset= -32 (504) */
/* 538 */
                                0x1b, /* FC_CARRAY */
                                0x3, /* 3 */
/* 540 */ NdrFcShort( 0x4 ), /* 4 */
/* 542 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                                0x0, /* */
/* 544 */ NdrFcShort( 0x0 ), /* 0 */
/* 546 */
                                0x4b, /* FC_PP */
                                0x5c, /* FC_PAD */
/* 548 */
                                0x48, /*
FC_VARIABLE_REPEAT */
                                0x49, /*
FC_FIXED_OFFSET */
/* 550 */ NdrFcShort( 0x4 ), /* 4 */
/* 552 */ NdrFcShort( 0x0 ), /* 0 */
/* 554 */ NdrFcShort( 0x1 ), /* 1 */
/* 556 */ NdrFcShort( 0x0 ), /* 0 */
/* 558 */ NdrFcShort( 0x0 ), /* 0 */
/* 560 */ 0x12, 0x0, /* FC_UP */
/* 562 */ NdrFcShort( 0x182 ), /* Offset= 386 (948) */
/* 564 */
                                0x5b, /* FC_END */
                                0x8, /* FC_LONG */
/* 566 */ 0x5c, /* FC_PAD */
                                0x5b, /* FC_END */
/* 568 */
                                0x1a, /*
FC_BOGUS_STRUCT */
                                0x3, /* 3 */
/* 570 */ NdrFcShort( 0x8 ), /* 8 */
/* 572 */ NdrFcShort( 0x0 ), /* 0 */
/* 574 */ NdrFcShort( 0x6 ), /* Offset= 6 (580) */
/* 576 */ 0x8, /* FC_LONG */
                                0x36, /* FC_POINTER */
/* 578 */ 0x5c, /* FC_PAD */
                                0x5b, /* FC_END */
/* 580 */
                                0x11, 0x0, /* FC_RP */
/* 582 */ NdrFcShort( 0xfffffd4 ), /* Offset= -44 (538) */
/* 584 */
                                0x2f, /* FC_IP */
                                0x5a, /*
FC_CONSTANT_IID */
/* 586 */ NdrFcLong( 0x2f ), /* 47 */
/* 590 */ NdrFcShort( 0x0 ), /* 0 */
/* 592 */ NdrFcShort( 0x0 ), /* 0 */
/* 594 */ 0xc0, /* 192 */
                                0x0, /* 0 */
/* 596 */ 0x0, /* 0 */
                                0x0, /* 0 */
/* 598 */ 0x0, /* 0 */
                                0x0, /* 0 */
/* 600 */ 0x0, /* 0 */
                                0x46, /* 70 */
/* 602 */
                                0x1b, /* FC_CARRAY */
                                0x0, /* 0 */
/* 604 */ NdrFcShort( 0x1 ), /* 1 */
/* 606 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                                0x0, /* */
/* 608 */ NdrFcShort( 0x4 ), /* 4 */
/* 610 */ 0x1, /* FC_BYTE */
                                0x5b, /* FC_END */
                                0x1a, /*
FC_VARIABLE_REPEAT */
                                0x48, /*
FC_FIXED_OFFSET */
/* 614 */ NdrFcShort( 0x10 ), /* 16 */
/* 616 */ NdrFcShort( 0x0 ), /* 0 */
/* 618 */ NdrFcShort( 0xa ), /* Offset= 10 (628) */
/* 620 */ 0x8, /* FC_LONG */
                                0x8, /* FC_LONG */
/* 622 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
                                0x0, /* 0 */
/* 624 */ NdrFcShort( 0xfffffd8 ), /* Offset= -40 (584) */
/* 626 */ 0x36, /* FC_POINTER */
                                0x5b, /* FC_END */
/* 628 */
                                0x12, 0x0, /* FC_UP */
/* 630 */ NdrFcShort( 0xfffffe4 ), /* Offset= -28 (602) */
/* 632 */
                                0x1b, /* FC_CARRAY */
                                0x3, /* 3 */
/* 634 */ NdrFcShort( 0x4 ), /* 4 */
/* 636 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                                0x0, /* */
/* 638 */ NdrFcShort( 0x0 ), /* 0 */
/* 640 */
                                0x4b, /* FC_PP */
                                0x5c, /* FC_PAD */
/* 642 */
                                0x48, /*
FC_VARIABLE_REPEAT */
                                0x49, /*
FC_FIXED_OFFSET */
/* 644 */ NdrFcShort( 0x4 ), /* 4 */
/* 646 */ NdrFcShort( 0x0 ), /* 0 */
/* 648 */ NdrFcShort( 0x1 ), /* 1 */
/* 650 */ NdrFcShort( 0x0 ), /* 0 */
/* 652 */ NdrFcShort( 0x0 ), /* 0 */
/* 654 */ 0x12, 0x0, /* FC_UP */
/* 656 */ NdrFcShort( 0xfffffd4 ), /* Offset= -44 (612) */
/* 658 */
                                0x5b, /* FC_END */
                                0x8, /* FC_LONG */
/* 660 */ 0x5c, /* FC_PAD */
                                0x5b, /* FC_END */
/* 662 */
                                0x1a, /*
FC_BOGUS_STRUCT */
                                0x3, /* 3 */
/* 664 */ NdrFcShort( 0x8 ), /* 8 */
/* 666 */ NdrFcShort( 0x0 ), /* 0 */
/* 668 */ NdrFcShort( 0x6 ), /* Offset= 6 (674) */
/* 670 */ 0x8, /* FC_LONG */
                                0x36, /* FC_POINTER */
/* 672 */ 0x5c, /* FC_PAD */
                                0x5b, /* FC_END */
/* 674 */
                                0x11, 0x0, /* FC_RP */
/* 676 */ NdrFcShort( 0xfffffd4 ), /* Offset= -44 (632) */
/* 678 */
                                0x1d, /* FC_SMFARRAY */
                                0x0, /* 0 */
/* 680 */ NdrFcShort( 0x8 ), /* 8 */
/* 682 */ 0x2, /* FC_CHAR */
                                0x5b, /* FC_END */
/* 684 */
                                0x15, /* FC_STRUCT */

```

/* 686 */ NdrFcShort(0x10),	0x3,	/* 3 */	/* 758 */ NdrFcShort(0x8),	0x3,	/* 3 */
/* 688 */ 0x8,	/* 16 */		/* 760 */	/* 8 */	
	/* FC_LONG */				
/* 690 */ 0x6,	0x6,	/* FC_SHORT */		0x4b,	/* FC_PP */
	/* FC_SHORT */			0x5c,	/* FC_PAD */
	0x4c,	/*	/* 762 */		
FC_EMBEDDED_COMPLEX */				0x46,	/* FC_NO_REPEAT */
/* 692 */ 0x0,	/* 0 */			0x5c,	/* FC_PAD */
	NdrFcShort(0xffffffff),	/* Offset=	/* 764 */ NdrFcShort(0x4),	/* 4 */	
-15 (678) */			/* 766 */ NdrFcShort(0x4),	/* 4 */	
	0x5b,	/* FC_END */	/* 768 */ 0x12, 0x0, /* FC_UP */		
/* 696 */			/* 770 */ NdrFcShort(0xffffffffe8),	/* Offset= -24 (746) */	
	0x1a,	/*	/* 772 */		
FC_BOGUS_STRUCT */				0x5b,	/* FC_END */
	0x3,	/* 3 */			
/* 698 */ NdrFcShort(0x18),	/* 24 */		/* 774 */ 0x8,	0x8,	/* FC_LONG */
/* 700 */ NdrFcShort(0x0),	/* 0 */			/* FC_LONG */	
/* 702 */ NdrFcShort(0xa),	/* Offset= 10 (712) */			0x5b,	/* FC_END */
/* 704 */ 0x8,	/* FC_LONG */		/* 776 */		
	0x36,	/* FC_POINTER */		0x1b,	/* FC_CARRAY */
/* 706 */ 0x4c,	/* FC_EMBEDDED_COMPLEX */			0x3,	/* 3 */
	0x0,	/* 0 */	/* 778 */ NdrFcShort(0x4),	/* 4 */	
/* 708 */ NdrFcShort(0xffffffffe8),	/* Offset= -24 (684) */		/* 780 */ 0x19,	/* Corr desc: field pointer, FC_ULONG */	
/* 710 */ 0x5c,	/* FC_PAD */			0x0,	/* */
	0x5b,	/* FC_END */	/* 782 */ NdrFcShort(0x0),	/* 0 */	
/* 712 */			/* 784 */ 0x8,	/* FC_LONG */	
	0x11, 0x0, /* FC_RP */			0x5b,	/* FC_END */
/* 714 */ NdrFcShort(0xffffffff0c),	/* Offset= -244 (470) */		/* 786 */		
/* 716 */				0x16,	/* FC_PSTRUCT */
	0x1b,	/* FC_CARRAY */		0x3,	/* 3 */
	0x0,	/* 0 */	/* 788 */ NdrFcShort(0x8),	/* 8 */	
/* 718 */ NdrFcShort(0x1),	/* 1 */		/* 790 */		
/* 720 */ 0x19,	/* Corr desc: field pointer, FC_ULONG */			0x4b,	/* FC_PP */
	0x0,	/* */		0x5c,	/* FC_PAD */
/* 722 */ NdrFcShort(0x0),	/* 0 */		/* 792 */		
/* 724 */ 0x1,	/* FC_BYTE */			0x46,	/* FC_NO_REPEAT */
	0x5b,	/* FC_END */		0x5c,	/* FC_PAD */
/* 726 */			/* 794 */ NdrFcShort(0x4),	/* 4 */	
	0x16,	/* FC_PSTRUCT */	/* 796 */ NdrFcShort(0x4),	/* 4 */	
	0x3,	/* 3 */	/* 798 */ 0x12, 0x0, /* FC_UP */		
/* 728 */ NdrFcShort(0x8),	/* 8 */		/* 800 */ NdrFcShort(0xffffffffe8),	/* Offset= -24 (776) */	
/* 730 */			/* 802 */		
	0x4b,	/* FC_PP */		0x5b,	/* FC_END */
	0x5c,	/* FC_PAD */			
/* 732 */				0x8,	/* FC_LONG */
	0x46,	/* FC_NO_REPEAT */	/* 804 */ 0x8,	/* FC_LONG */	
	0x5c,	/* FC_PAD */		0x5b,	/* FC_END */
/* 734 */ NdrFcShort(0x4),	/* 4 */		/* 806 */		
/* 736 */ NdrFcShort(0x4),	/* 4 */			0x1b,	/* FC_CARRAY */
/* 738 */ 0x12, 0x0, /* FC_UP */				0x7,	/* 7 */
/* 740 */ NdrFcShort(0xffffffffe8),	/* Offset= -24 (716) */		/* 808 */ NdrFcShort(0x8),	/* 8 */	
/* 742 */			/* 810 */ 0x19,	/* Corr desc: field pointer, FC_ULONG */	
	0x5b,	/* FC_END */		0x0,	/* */
	0x8,	/* FC_LONG */	/* 812 */ NdrFcShort(0x0),	/* 0 */	
/* 744 */ 0x8,	/* FC_LONG */		/* 814 */ 0xb,	/* FC_HYPER */	
	0x5b,	/* FC_END */		0x5b,	/* FC_END */
/* 746 */			/* 816 */		
	0x1b,	/* FC_CARRAY */		0x16,	/* FC_PSTRUCT */
	0x1,	/* 1 */		0x3,	/* 3 */
/* 748 */ NdrFcShort(0x2),	/* 2 */		/* 818 */ NdrFcShort(0x8),	/* 8 */	
/* 750 */ 0x19,	/* Corr desc: field pointer, FC_ULONG */		/* 820 */		
	0x0,	/* */		0x4b,	/* FC_PP */
/* 752 */ NdrFcShort(0x0),	/* 0 */			0x5c,	/* FC_PAD */
/* 754 */ 0x6,	/* FC_SHORT */		/* 822 */		
	0x5b,	/* FC_END */		0x46,	/* FC_NO_REPEAT */
/* 756 */				0x5c,	/* FC_PAD */
	0x16,	/* FC_PSTRUCT */	/* 824 */ NdrFcShort(0x4),	/* 4 */	
			/* 826 */ NdrFcShort(0x4),	/* 4 */	

/* 828 */	0x12, 0x0, /* FC_UP */				0x5c,	/* FC_PAD */
/* 830 */	NdrFcShort(0xffffffe8),	/* Offset= -24 (806) */		/* 900 */		
/* 832 */					0x12, 0x0, /* FC_UP */	
	0x5b,	/* FC_END */		/* 902 */	NdrFcShort(0xfffffd90),	/* Offset= -624 (278) */
				/* 904 */		
/* 834 */	0x8,	/* FC_LONG */			0x12, 0x10,	/* FC_UP
	/* FC_LONG */			[pointer_deref] */		
/* 836 */	0x5b,	/* FC_END */		/* 906 */	NdrFcShort(0xfffffd92),	/* Offset= -622 (284) */
				/* 908 */		
	0x15,	/* FC_STRUCT */			0x12, 0x10,	/* FC_UP
	0x3,	/* 3 */		[pointer_deref] */		
/* 838 */	NdrFcShort(0x8),	/* 8 */		/* 910 */	NdrFcShort(0xfffffda6),	/* Offset= -602 (308) */
/* 840 */	0x8,	/* FC_LONG */		/* 912 */		
	0x8,	/* FC_LONG */			0x12, 0x10,	/* FC_UP
/* 842 */	0x5c,	/* FC_PAD */		[pointer_deref] */		
	0x5b,	/* FC_END */		/* 914 */	NdrFcShort(0xfffffdb4),	/* Offset= -588 (326) */
/* 844 */				/* 916 */		
	0x1b,	/* FC_CARRAY */			0x12, 0x10,	/* FC_UP
	0x3,	/* 3 */		[pointer_deref] */		
/* 846 */	NdrFcShort(0x8),	/* 8 */		/* 918 */	NdrFcShort(0xfffffdc2),	/* Offset= -574 (344) */
/* 848 */	0x7,	/* Corr desc: FC_USHORT */		/* 920 */		
	0x0,	/* */			0x12, 0x10,	/* FC_UP
/* 850 */	NdrFcShort(0xffd8),	/* -40 */		[pointer_deref] */		
/* 852 */	0x4c,	/* FC_EMBEDDED_COMPLEX */		/* 922 */	NdrFcShort(0x2),	/* Offset= 2 (924) */
	0x0,	/* 0 */		/* 924 */		
/* 854 */	NdrFcShort(0xfffffee),	/* Offset= -18 (836) */			0x12, 0x0, /* FC_UP */	
/* 856 */	0x5c,	/* FC_PAD */		/* 926 */	NdrFcShort(0x16),	/* Offset= 22 (948) */
	0x5b,	/* FC_END */		/* 928 */		
/* 858 */					0x15,	/* FC_STRUCT */
	0x1a,	/*			0x7,	/* 7 */
FC_BOGUS_STRUCT */				/* 930 */	NdrFcShort(0x10),	/* 16 */
	0x3,	/* 3 */		/* 932 */	0x6,	/* FC_SHORT */
/* 860 */	NdrFcShort(0x28),	/* 40 */			0x1,	/* FC_BYTE */
/* 862 */	NdrFcShort(0xfffffee),	/* Offset= -18 (844) */		/* 934 */	0x1,	/* FC_BYTE */
/* 864 */	NdrFcShort(0x0),	/* Offset= 0 (864) */		/* 936 */	0x8,	/* FC_ALIGNM4 */
/* 866 */	0x6,	/* FC_SHORT */			0x38,	/* FC_LONG */
	0x6,	/* FC_SHORT */		/* 938 */	0xb,	/* FC_HYPER */
/* 868 */	0x38,	/* FC_ALIGNM4 */			0x5b,	/* FC_END */
	0x8,	/* FC_LONG */		/* 940 */		
/* 870 */	0x8,	/* FC_LONG */			0x12, 0x0, /* FC_UP */	
	0x4c,	/*		/* 942 */	NdrFcShort(0xffffff2),	/* Offset= -14 (928) */
FC_EMBEDDED_COMPLEX */				/* 944 */		
/* 872 */	0x0,	/* 0 */			0x12, 0x8, /* FC_UP [simple_pointer] */	
		NdrFcShort(0xfffffd7),	/* Offset=	/* 946 */	0x2,	/* FC_CHAR */
-521 (352) */					0x5c,	/* FC_PAD */
/* 876 */	0x5b,	/* FC_END */		/* 948 */		
					0x1a,	/*
	0x12, 0x0, /* FC_UP */			FC_BOGUS_STRUCT */		
/* 878 */	NdrFcShort(0xfffffef6),	/* Offset= -266 (612) */			0x7,	/* 7 */
/* 880 */				/* 950 */	NdrFcShort(0x20),	/* 32 */
	0x12, 0x8, /* FC_UP [simple_pointer] */			/* 952 */	NdrFcShort(0x0),	/* 0 */
/* 882 */	0x1,	/* FC_BYTE */		/* 954 */	NdrFcShort(0x0),	/* Offset= 0 (954) */
	0x5c,	/* FC_PAD */		/* 956 */	0x8,	/* FC_LONG */
/* 884 */					0x8,	/* FC_LONG */
	0x12, 0x8, /* FC_UP [simple_pointer] */			/* 958 */	0x6,	/* FC_SHORT */
/* 886 */	0x6,	/* FC_SHORT */			0x6,	/* FC_SHORT */
	0x5c,	/* FC_PAD */		/* 960 */	0x6,	/* FC_SHORT */
/* 888 */					0x6,	/* FC_SHORT */
	0x12, 0x8, /* FC_UP [simple_pointer] */			/* 962 */	0x4c,	/* FC_EMBEDDED_COMPLEX */
/* 890 */	0x8,	/* FC_LONG */			0x0,	/* 0 */
	0x5c,	/* FC_PAD */		/* 964 */	NdrFcShort(0xfffffc42),	/* Offset= -958 (6) */
/* 892 */				/* 966 */	0x5c,	/* FC_PAD */
	0x12, 0x8, /* FC_UP [simple_pointer] */				0x5b,	/* FC_END */
/* 894 */	0xa,	/* FC_FLOAT */		/* 968 */	0xb4,	/* FC_USER_MARSHAL */
	0x5c,	/* FC_PAD */			0x83,	/* 131 */
/* 896 */				/* 970 */	NdrFcShort(0x0),	/* 0 */
	0x12, 0x8, /* FC_UP [simple_pointer] */			/* 972 */	NdrFcShort(0x10),	/* 16 */
/* 898 */	0xc,	/* FC_DOUBLE */				

```

/* 974 */ NdrFcShort( 0x0 ), /* 0 */
/* 976 */ NdrFcShort( 0xfffffc32 ), /* Offset= -974 (2) */
/* 978 */
                                0x11, 0x4, /* FC_RP [allocated_on_stack] */
/* 980 */ NdrFcShort( 0x6 ), /* Offset= 6 (986) */
/* 982 */
                                0x13, 0x0, /* FC_OP */
/* 984 */ NdrFcShort( 0xfffffdc ), /* Offset= -36 (948) */
/* 986 */ 0xb4, /* FC_USER_MARSHAL */
                                0x83, /* 131 */
/* 988 */ NdrFcShort( 0x0 ), /* 0 */
/* 990 */ NdrFcShort( 0x10 ), /* 16 */
/* 992 */ NdrFcShort( 0x0 ), /* 0 */
/* 994 */ NdrFcShort( 0xfffffff4 ), /* Offset= -12 (982) */

                                0x0

    }
};

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

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

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

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

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

    return 0;
}

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

```

```

#endif /* !defined(_M_IA64) && !defined(_M_AXP64) */

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

/* this ALWAYS GENERATED file contains the proxy stub code */

/* File created by MIDL compiler version 5.03.0280 */
/* at Sat Apr 08 16:40:10 2000
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext,
    robust
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
    DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADING( )

#ifdef _M_IA64 || defined(_M_AXP64)
#define USE_STUBLESS_PROXY

/* verify that the <rpcproxy.h> version is high enough to compile this file*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 475
#endif

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

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 979
#define PROC_FORMAT_STRING_SIZE 253
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 1

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

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

extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */

```



```

#else
                                NdrFcShort( 0x30 ), /* axp64 Stack
size/offset = 48 */
#endif
/* 10 */ NdrFcShort( 0x0 ), /* 0 */
/* 12 */ NdrFcShort( 0x8 ), /* 8 */
/* 14 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
                                0x3, /* 3 */
/* 16 */ 0xa, /* 10 */
                                0x7, /* Ext Flags: new corr
desc, clt corr check, srv corr check, */
/* 18 */ NdrFcShort( 0x20 ), /* 32 */
/* 20 */ NdrFcShort( 0x20 ), /* 32 */
/* 22 */ NdrFcShort( 0x0 ), /* 0 */
/* 24 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 26 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifndef _ALPHA_
/* 28 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
                                NdrFcShort( 0x8 ), /* axp64 Stack
size/offset = 8 */
#endif
/* 30 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

/* Parameter txn_out */

/* 32 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out,
simple ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 34 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
                                NdrFcShort( 0x20 ), /* axp64 Stack
size/offset = 32 */
#endif
/* 36 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Return value */

/* 38 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
/* 40 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
                                NdrFcShort( 0x28 ), /* axp64 Stack
size/offset = 40 */
#endif
/* 42 */ 0x8, /* FC_LONG */
                                0x0, /* 0 */

/* Procedure Payment */

/* 44 */ 0x33, /* FC_AUTO_HANDLE */
                                0x6c, /* Old Flags: object,
Oi2 */
/* 46 */ NdrFcLong( 0x0 ), /* 0 */
/* 50 */ NdrFcShort( 0x4 ), /* 4 */
#ifndef _ALPHA_
/* 52 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
                                NdrFcShort( 0x30 ), /* axp64 Stack
size/offset = 48 */
#endif
/* 54 */ NdrFcShort( 0x0 ), /* 0 */
/* 56 */ NdrFcShort( 0x8 ), /* 8 */

/* 58 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
                                0x3, /* 3 */
/* 60 */ 0xa, /* 10 */
                                0x7, /* Ext Flags: new corr
desc, clt corr check, srv corr check, */
/* 62 */ NdrFcShort( 0x20 ), /* 32 */
/* 64 */ NdrFcShort( 0x20 ), /* 32 */
/* 66 */ NdrFcShort( 0x0 ), /* 0 */
/* 68 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 70 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifndef _ALPHA_
/* 72 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
                                NdrFcShort( 0x8 ), /* axp64 Stack
size/offset = 8 */
#endif
/* 74 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

/* Parameter txn_out */

/* 76 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out,
simple ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 78 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
                                NdrFcShort( 0x20 ), /* axp64 Stack
size/offset = 32 */
#endif
/* 80 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Return value */

/* 82 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
/* 84 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
                                NdrFcShort( 0x28 ), /* axp64 Stack
size/offset = 40 */
#endif
/* 86 */ 0x8, /* FC_LONG */
                                0x0, /* 0 */

/* Procedure Delivery */

/* 88 */ 0x33, /* FC_AUTO_HANDLE */
                                0x6c, /* Old Flags: object,
Oi2 */
/* 90 */ NdrFcLong( 0x0 ), /* 0 */
/* 94 */ NdrFcShort( 0x5 ), /* 5 */
#ifndef _ALPHA_
/* 96 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
                                NdrFcShort( 0x30 ), /* axp64 Stack
size/offset = 48 */
#endif
/* 98 */ NdrFcShort( 0x0 ), /* 0 */
/* 100 */ NdrFcShort( 0x8 ), /* 8 */
/* 102 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
                                0x3, /* 3 */
/* 104 */ 0xa, /* 10 */
                                0x7, /* Ext Flags: new corr
desc, clt corr check, srv corr check, */
/* 106 */ NdrFcShort( 0x20 ), /* 32 */

```

```

/* 108 */ NdrFcShort( 0x20 ), /* 32 */
/* 110 */ NdrFcShort( 0x0 ), /* 0 */
/* 112 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 114 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef _ALPHA_
/* 116 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
NdrFcShort( 0x8 ), /* axp64 Stack
size/offset = 8 */
#endif
/* 118 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

/* Parameter txn_out */

/* 120 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out,
simple ref, srv alloc size=24 */
#ifdef _ALPHA_
/* 122 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
NdrFcShort( 0x20 ), /* axp64 Stack
size/offset = 32 */
#endif
/* 124 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Return value */

/* 126 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA_
/* 128 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
NdrFcShort( 0x28 ), /* axp64 Stack
size/offset = 40 */
#endif
/* 130 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure StockLevel */

/* 132 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */
/* 134 */ NdrFcLong( 0x0 ), /* 0 */
/* 138 */ NdrFcShort( 0x6 ), /* 6 */
#ifdef _ALPHA_
/* 140 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
NdrFcShort( 0x30 ), /* axp64 Stack
size/offset = 48 */
#endif
/* 142 */ NdrFcShort( 0x0 ), /* 0 */
/* 144 */ NdrFcShort( 0x8 ), /* 8 */
/* 146 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
0x3, /* 3 */
/* 148 */ 0xa, /* 10 */
0x7, /* Ext Flags: new corr
desc, clt corr check, srv corr check, */
/* 150 */ NdrFcShort( 0x20 ), /* 32 */
/* 152 */ NdrFcShort( 0x20 ), /* 32 */
/* 154 */ NdrFcShort( 0x0 ), /* 0 */
/* 156 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 158 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef _ALPHA_
/* 160 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
NdrFcShort( 0x8 ), /* axp64 Stack
size/offset = 8 */
#endif
/* 162 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

/* Parameter txn_out */

/* 164 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out,
simple ref, srv alloc size=24 */
#ifdef _ALPHA_
/* 166 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
NdrFcShort( 0x20 ), /* axp64 Stack
size/offset = 32 */
#endif
/* 168 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Return value */

/* 170 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA_
/* 172 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
NdrFcShort( 0x28 ), /* axp64 Stack
size/offset = 40 */
#endif
/* 174 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure OrderStatus */

/* 176 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */
/* 178 */ NdrFcLong( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x7 ), /* 7 */
#ifdef _ALPHA_
/* 184 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
NdrFcShort( 0x30 ), /* axp64 Stack
size/offset = 48 */
#endif
/* 186 */ NdrFcShort( 0x0 ), /* 0 */
/* 188 */ NdrFcShort( 0x8 ), /* 8 */
/* 190 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
0x3, /* 3 */
/* 192 */ 0xa, /* 10 */
0x7, /* Ext Flags: new corr
desc, clt corr check, srv corr check, */
/* 194 */ NdrFcShort( 0x20 ), /* 32 */
/* 196 */ NdrFcShort( 0x20 ), /* 32 */
/* 198 */ NdrFcShort( 0x0 ), /* 0 */
/* 200 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 202 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef _ALPHA_
/* 204 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
NdrFcShort( 0x8 ), /* axp64 Stack
size/offset = 8 */
#endif
/* 206 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

```

```

/* Parameter txn_out */

/* 208 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out,
simple ref, srv alloc size=24 */
#ifdef _ALPHA_
/* 210 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
NdrFcShort( 0x20 ), /* xpp64 Stack
size/offset = 32 */
#endif
/* 212 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Return value */

/* 214 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA_
/* 216 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
NdrFcShort( 0x28 ), /* xpp64 Stack
size/offset = 40 */
#endif
/* 218 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure CallSetComplete */

/* 220 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */
/* 222 */ NdrFcLong( 0x0 ), /* 0 */
/* 226 */ NdrFcShort( 0x8 ), /* 8 */
/* 228 */ NdrFcShort( 0x10 ), /* ia64, xpp64 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, xpp64 Stack size/offset = 8 */
/* 250 */ 0x8, /* FC_LONG */
0x0, /* 0 */
0x0

}
};

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
0,
{
NdrFcShort( 0x0 ), /* 0 */
/* 2 */
0x12, 0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0x39e ), /* Offset= 926 (930) */
/* 6 */
0x2b, /*
FC_NON_ENCAPSULATED_UNION */
0x9, /* FC_ULONG */
/* 8 */ 0x7, /* Corr desc: FC_USHORT */
0x0, /* */
/* 10 */ NdrFcShort( 0xffff8 ), /* -8 */
/* 12 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 14 */ NdrFcShort( 0x2 ), /* Offset= 2 (16) */
/* 16 */ NdrFcShort( 0x10 ), /* 16 */
/* 18 */ NdrFcShort( 0x2b ), /* 43 */
/* 20 */ NdrFcLong( 0x3 ), /* 3 */
/* 24 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 26 */ NdrFcLong( 0x11 ), /* 17 */
/* 30 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE */
/* 32 */ NdrFcLong( 0x2 ), /* 2 */
/* 36 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 38 */ NdrFcLong( 0x4 ), /* 4 */
/* 42 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT */
/* 44 */ NdrFcLong( 0x5 ), /* 5 */
/* 48 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 50 */ NdrFcLong( 0xb ), /* 11 */
/* 54 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 56 */ NdrFcLong( 0xa ), /* 10 */
/* 60 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 62 */ NdrFcLong( 0x6 ), /* 6 */
/* 66 */ NdrFcShort( 0xd6 ), /* Offset= 214 (280) */
/* 68 */ NdrFcLong( 0x7 ), /* 7 */
/* 72 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 74 */ NdrFcLong( 0x8 ), /* 8 */
/* 78 */ NdrFcShort( 0xd0 ), /* Offset= 208 (286) */
/* 80 */ NdrFcLong( 0xd ), /* 13 */
/* 84 */ NdrFcShort( 0xe4 ), /* Offset= 228 (312) */
/* 86 */ NdrFcLong( 0x9 ), /* 9 */
/* 90 */ NdrFcShort( 0xf0 ), /* Offset= 240 (330) */
/* 92 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 96 */ NdrFcShort( 0xfc ), /* Offset= 252 (348) */
/* 98 */ NdrFcLong( 0x24 ), /* 36 */
/* 102 */ NdrFcShort( 0x2f4 ), /* Offset= 756 (858) */
/* 104 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 108 */ NdrFcShort( 0x2ee ), /* Offset= 750 (858) */
/* 110 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 114 */ NdrFcShort( 0x2ec ), /* Offset= 748 (862) */
/* 116 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 120 */ NdrFcShort( 0x2ea ), /* Offset= 746 (866) */
/* 122 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 126 */ NdrFcShort( 0x2e8 ), /* Offset= 744 (870) */
/* 128 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 132 */ NdrFcShort( 0x2e6 ), /* Offset= 742 (874) */
/* 134 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 138 */ NdrFcShort( 0x2e4 ), /* Offset= 740 (878) */
/* 140 */ NdrFcLong( 0x400b ), /* 16395 */
/* 144 */ NdrFcShort( 0x2d2 ), /* Offset= 722 (866) */
/* 146 */ NdrFcLong( 0x400a ), /* 16394 */
/* 150 */ NdrFcShort( 0x2d0 ), /* Offset= 720 (870) */
/* 152 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 156 */ NdrFcShort( 0x2d6 ), /* Offset= 726 (882) */
/* 158 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 162 */ NdrFcShort( 0x2cc ), /* Offset= 716 (878) */
/* 164 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 168 */ NdrFcShort( 0x2ce ), /* Offset= 718 (886) */
/* 170 */ NdrFcLong( 0x400d ), /* 16397 */
/* 174 */ NdrFcShort( 0x2cc ), /* Offset= 716 (890) */
/* 176 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 180 */ NdrFcShort( 0x2ca ), /* Offset= 714 (894) */
/* 182 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 186 */ NdrFcShort( 0x2c8 ), /* Offset= 712 (898) */
/* 188 */ NdrFcLong( 0x400c ), /* 16396 */
/* 192 */ NdrFcShort( 0x2c6 ), /* Offset= 710 (902) */
/* 194 */ NdrFcLong( 0x10 ), /* 16 */

```

/* 198 */ NdrFcShort(0x8002),	/* Simple arm type: FC_CHAR */	0x0,	/* 0 */
/* 200 */ NdrFcLong(0x12),	/* 18 */	/* 0 */	
/* 204 */ NdrFcShort(0x8006),	/* Simple arm type: FC_SHORT */	0x46,	/* 70 */
/* 206 */ NdrFcLong(0x13),	/* 19 */		
/* 210 */ NdrFcShort(0x8008),	/* Simple arm type: FC_LONG */	0x2f,	/* FC_IP */
/* 212 */ NdrFcLong(0x16),	/* 22 */	0x5a,	/*
/* 216 */ NdrFcShort(0x8008),	/* Simple arm type: FC_LONG */	FC_CONSTANT_IID */	
/* 218 */ NdrFcLong(0x17),	/* 23 */	/* 332 */ NdrFcLong(0x20400),	/* 132096 */
/* 222 */ NdrFcShort(0x8008),	/* Simple arm type: FC_LONG */	/* 336 */ NdrFcShort(0x0),	/* 0 */
/* 224 */ NdrFcLong(0xe),	/* 14 */	/* 338 */ NdrFcShort(0x0),	/* 0 */
/* 228 */ NdrFcShort(0x2aa),	/* Offset= 682 (910) */	/* 340 */ 0xc0,	/* 192 */
/* 230 */ NdrFcLong(0x400e),	/* 16398 */	0x0,	/* 0 */
/* 234 */ NdrFcShort(0x2b0),	/* Offset= 688 (922) */	/* 342 */ 0x0,	/* 0 */
/* 236 */ NdrFcLong(0x4010),	/* 16400 */	0x0,	/* 0 */
/* 240 */ NdrFcShort(0x2ae),	/* Offset= 686 (926) */	/* 344 */ 0x0,	/* 0 */
/* 242 */ NdrFcLong(0x4012),	/* 16402 */	0x0,	/* 0 */
/* 246 */ NdrFcShort(0x26c),	/* Offset= 620 (866) */	/* 346 */ 0x0,	/* 0 */
/* 248 */ NdrFcLong(0x4013),	/* 16403 */	0x46,	/* 70 */
/* 252 */ NdrFcShort(0x26a),	/* Offset= 618 (870) */		
/* 254 */ NdrFcLong(0x4016),	/* 16406 */	0x12, 0x10,	/* FC_UP
/* 258 */ NdrFcShort(0x264),	/* Offset= 612 (870) */	[pointer_deref] */	
/* 260 */ NdrFcLong(0x4017),	/* 16407 */	/* 350 */ NdrFcShort(0x2),	/* Offset= 2 (352) */
/* 264 */ NdrFcShort(0x25e),	/* Offset= 606 (870) */	/* 352 */	
/* 266 */ NdrFcLong(0x0),	/* 0 */	0x12, 0x0,	/* FC_UP */
/* 270 */ NdrFcShort(0x0),	/* Offset= 0 (270) */	/* 354 */ NdrFcShort(0x1e6),	/* Offset= 486 (840) */
/* 272 */ NdrFcLong(0x1),	/* 1 */	/* 356 */	
/* 276 */ NdrFcShort(0x0),	/* Offset= 0 (276) */	0x2a,	/*
/* 278 */ NdrFcShort(0xffffffff),	/* Offset= -1 (277) */	FC_ENCAPSULATED_UNION */	
/* 280 */		0x89,	/* 137 */
	0x15,	/* FC_STRUCT */	
	0x7,	/* 7 */	
/* 282 */ NdrFcShort(0x8),	/* 8 */	/* 358 */ NdrFcShort(0x20),	/* 32 */
/* 284 */ 0xb,	/* FC_HYPER */	/* 360 */ NdrFcShort(0xa),	/* 10 */
	0x5b,	/* 362 */ NdrFcLong(0x8),	/* 8 */
/* 286 */		/* 366 */ NdrFcShort(0x50),	/* Offset= 80 (446) */
	0x12, 0x0,	/* 368 */ NdrFcLong(0xd),	/* 13 */
/* 288 */ NdrFcShort(0xe),	/* Offset= 14 (302) */	/* 372 */ NdrFcShort(0x70),	/* Offset= 112 (484) */
/* 290 */		/* 374 */ NdrFcLong(0x9),	/* 9 */
	0x1b,	/* 378 */ NdrFcShort(0x90),	/* Offset= 144 (522) */
/* 292 */ NdrFcShort(0x2),	/* 2 */	/* 380 */ NdrFcLong(0xc),	/* 12 */
/* 294 */ 0x9,	/* Corr desc: FC_ULONG */	/* 384 */ NdrFcShort(0xb0),	/* Offset= 176 (560) */
	0x0,	/* 386 */ NdrFcLong(0x24),	/* 36 */
/* 296 */ NdrFcShort(0xffffc),	/* -4 */	/* 390 */ NdrFcShort(0x104),	/* Offset= 260 (650) */
/* 298 */ NdrFcShort(0x1),	/* Corr flags: early, */	/* 392 */ NdrFcLong(0x800d),	/* 32781 */
/* 300 */ 0x6,	/* FC_SHORT */	/* 396 */ NdrFcShort(0x120),	/* Offset= 288 (684) */
	0x5b,	/* 398 */ NdrFcLong(0x10),	/* 16 */
/* 302 */		/* 402 */ NdrFcShort(0x13a),	/* Offset= 314 (716) */
	0x17,	/* 404 */ NdrFcLong(0x2),	/* 2 */
/* 304 */ NdrFcShort(0x8),	/* 8 */	/* 408 */ NdrFcShort(0x150),	/* Offset= 336 (744) */
/* 306 */ NdrFcShort(0xffffffff0),	/* Offset= -16 (290) */	/* 410 */ NdrFcLong(0x3),	/* 3 */
/* 308 */ 0x8,	/* FC_LONG */	/* 414 */ NdrFcShort(0x166),	/* Offset= 358 (772) */
	0x8,	/* 416 */ NdrFcLong(0x14),	/* 20 */
/* 310 */ 0x5c,	/* FC_PAD */	/* 420 */ NdrFcShort(0x17c),	/* Offset= 380 (800) */
	0x5b,	/* 422 */ NdrFcShort(0xffffffff),	/* Offset= -1 (421) */
/* 312 */		/* 424 */	
	0x2f,	0x21,	/*
FC_CONSTANT_IID */			
/* 314 */ NdrFcLong(0x0),	/* 0 */	FC_BOGUS_ARRAY */	
/* 318 */ NdrFcShort(0x0),	/* 0 */	0x3,	/* 3 */
/* 320 */ NdrFcShort(0x0),	/* 0 */	/* 426 */ NdrFcShort(0x0),	/* 0 */
/* 322 */ 0xc0,	/* 192 */	/* 428 */ 0x19,	/* Corr desc: field pointer, FC_ULONG */
	0x0,	0x0,	/* */
/* 324 */ 0x0,	/* 0 */	/* 430 */ NdrFcShort(0x0),	/* 0 */
	0x0,	/* 432 */ NdrFcShort(0x1),	/* Corr flags: early, */
/* 326 */ 0x0,	/* 0 */	/* 434 */ NdrFcLong(0xffffffff),	/* -1 */
		/* 438 */ NdrFcShort(0x0),	/* Corr flags: */
		/* 440 */	
		0x12, 0x0,	/* FC_UP */
		/* 442 */ NdrFcShort(0xffffffff74),	/* Offset= -140 (302) */
		/* 444 */ 0x5c,	/* FC_PAD */
		0x5b,	/* FC_END */

```

/* 446 */
FC_BOGUS_STRUCT */
    0x1a, /*
/* 448 */ NdrFcShort( 0x10 ), /* 16 */
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */ NdrFcShort( 0x6 ), /* Offset= 6 (458) */
/* 454 */ 0x8, /* FC_LONG */
/* 456 */ 0x36, /* FC_POINTER */
/* 458 */ 0x5b, /* FC_END */
    0x11, 0x0, /* FC_RP */
/* 460 */ NdrFcShort( 0xfffff5dc ), /* Offset= -36 (424) */
/* 462 */
FC_BOGUS_ARRAY */
    0x3, /* 3 */
/* 464 */ NdrFcShort( 0x0 ), /* 0 */
/* 466 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
    0x0, /*
/* 468 */ NdrFcShort( 0x0 ), /* 0 */
/* 470 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 472 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 476 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 478 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
    0x0, /* 0 */
/* 480 */ NdrFcShort( 0xfffff58 ), /* Offset= -168 (312) */
/* 482 */ 0x5c, /* FC_PAD */
    0x5b, /* FC_END */
/* 484 */
FC_BOGUS_STRUCT */
    0x1a, /*
/* 486 */ NdrFcShort( 0x10 ), /* 16 */
/* 488 */ NdrFcShort( 0x0 ), /* 0 */
/* 490 */ NdrFcShort( 0x6 ), /* Offset= 6 (496) */
/* 492 */ 0x8, /* FC_LONG */
    0x39, /* FC_ALIGNM8 */
/* 494 */ 0x36, /* FC_POINTER */
    0x5b, /* FC_END */
/* 496 */
    0x11, 0x0, /* FC_RP */
/* 498 */ NdrFcShort( 0xfffff5dc ), /* Offset= -36 (462) */
/* 500 */
    0x21, /*
FC_BOGUS_ARRAY */
    0x3, /* 3 */
/* 502 */ NdrFcShort( 0x0 ), /* 0 */
/* 504 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
    0x0, /*
/* 506 */ NdrFcShort( 0x0 ), /* 0 */
/* 508 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 510 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 514 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 516 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
    0x0, /* 0 */
/* 518 */ NdrFcShort( 0xfffff44 ), /* Offset= -188 (330) */
/* 520 */ 0x5c, /* FC_PAD */
    0x5b, /* FC_END */
/* 522 */
FC_BOGUS_STRUCT */
    0x1a, /*
/* 524 */ NdrFcShort( 0x10 ), /* 16 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 530 */ 0x8, /* FC_LONG */
    0x39, /* FC_ALIGNM8 */
/* 532 */ 0x36, /* FC_POINTER */
    0x5b, /* FC_END */
    0x11, 0x0, /* FC_RP */
/* 536 */ NdrFcShort( 0xfffff5dc ), /* Offset= -36 (500) */
/* 538 */
FC_BOGUS_ARRAY */
    0x3, /* 3 */
/* 540 */ NdrFcShort( 0x0 ), /* 0 */
/* 542 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
    0x0, /*
/* 544 */ NdrFcShort( 0x0 ), /* 0 */
/* 546 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 548 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 552 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 554 */
    0x12, 0x0, /* FC_UP */
/* 556 */ NdrFcShort( 0x176 ), /* Offset= 374 (930) */
/* 558 */ 0x5c, /* FC_PAD */
    0x5b, /* FC_END */
/* 560 */
    0x1a, /*
FC_BOGUS_STRUCT */
    0x3, /* 3 */
/* 562 */ NdrFcShort( 0x10 ), /* 16 */
/* 564 */ NdrFcShort( 0x0 ), /* 0 */
/* 566 */ NdrFcShort( 0x6 ), /* Offset= 6 (572) */
/* 568 */ 0x8, /* FC_LONG */
    0x39, /* FC_ALIGNM8 */
/* 570 */ 0x36, /* FC_POINTER */
    0x5b, /* FC_END */
/* 572 */
    0x11, 0x0, /* FC_RP */
/* 574 */ NdrFcShort( 0xfffff5dc ), /* Offset= -36 (538) */
/* 576 */
    0x2f, /* FC_IP */
    0x5a, /*
FC_CONSTANT_IID */
/* 578 */ NdrFcLong( 0x2f ), /* 47 */
/* 582 */ NdrFcShort( 0x0 ), /* 0 */
/* 584 */ NdrFcShort( 0x0 ), /* 0 */
/* 586 */ 0xc0, /* 192 */
    0x0, /* 0 */
/* 588 */ 0x0, /* 0 */
    0x0, /* 0 */
/* 590 */ 0x0, /* 0 */
    0x0, /* 0 */
/* 592 */ 0x0, /* 0 */
    0x46, /* 70 */
/* 594 */
    0x1b, /* FC_CARRAY */
    0x0, /* 0 */
/* 596 */ NdrFcShort( 0x1 ), /* 1 */
/* 598 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
    0x0, /*
/* 600 */ NdrFcShort( 0x4 ), /* 4 */
/* 602 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 604 */ 0x1, /* FC_BYTE */
    0x5b, /* FC_END */
/* 606 */
    0x1a, /*
FC_BOGUS_STRUCT */
    0x3, /* 3 */
/* 608 */ NdrFcShort( 0x18 ), /* 24 */
/* 610 */ NdrFcShort( 0x0 ), /* 0 */
/* 612 */ NdrFcShort( 0xc ), /* Offset= 12 (624) */

```

```

/* 614 */ 0x8, /* FC_LONG */
/* 616 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 618 */ NdrFcShort( 0xfffffd6 ), /* Offset= -42 (576) */
/* 620 */ 0x39, /* FC_ALIGNM8 */
/* 622 */ 0x5c, /* FC_POINTER */
/* 624 */ 0x5b, /* FC_END */
/* 626 */ 0x12, 0x0, /* FC_UP */
/* 628 */ NdrFcShort( 0xfffffe0 ), /* Offset= -32 (594) */
/* 630 */ 0x21, /*
FC_BOGUS_ARRAY */
/* 632 */ 0x3, /* 3 */
/* 634 */ NdrFcShort( 0x0 ), /* 0 */
/* 636 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 638 */ 0x0, /*
/* 640 */ NdrFcShort( 0x0 ), /* 0 */
/* 642 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 644 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 646 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 648 */ 0x12, 0x0, /* FC_UP */
/* 650 */ NdrFcShort( 0xfffffd8 ), /* Offset= -40 (606) */
/* 652 */ 0x5c, /* FC_PAD */
/* 654 */ 0x5b, /* FC_END */
/* 656 */ 0x1a, /*
FC_BOGUS_STRUCT */
/* 658 */ 0x3, /* 3 */
/* 660 */ NdrFcShort( 0x10 ), /* 16 */
/* 662 */ NdrFcShort( 0x0 ), /* 0 */
/* 664 */ NdrFcShort( 0x6 ), /* Offset= 6 (662) */
/* 666 */ 0x8, /* FC_LONG */
/* 668 */ 0x39, /* FC_ALIGNM8 */
/* 670 */ 0x36, /* FC_POINTER */
/* 672 */ 0x5b, /* FC_END */
/* 674 */ 0x11, 0x0, /* FC_RP */
/* 676 */ NdrFcShort( 0xfffffdc ), /* Offset= -36 (628) */
/* 678 */ 0x1d, /* FC_SMFARRAY */
/* 680 */ 0x0, /* 0 */
/* 682 */ NdrFcShort( 0x8 ), /* 8 */
/* 684 */ 0x2, /* FC_CHAR */
/* 686 */ 0x5b, /* FC_END */
/* 688 */ 0x15, /* FC_STRUCT */
/* 690 */ 0x3, /* 3 */
/* 692 */ NdrFcShort( 0x10 ), /* 16 */
/* 694 */ 0x8, /* FC_LONG */
/* 696 */ 0x6, /* FC_SHORT */
/* 698 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 700 */ 0x0, /* 0 */
/* 702 */ NdrFcShort( 0xffffffff1 ), /* Offset=
-15 (666) */
/* 704 */ 0x5b, /* FC_END */
/* 706 */ 0x1a, /*
FC_BOGUS_STRUCT */
/* 708 */ 0x3, /* 3 */
/* 710 */ NdrFcShort( 0x20 ), /* 32 */
/* 712 */ NdrFcShort( 0x0 ), /* 0 */
/* 714 */ NdrFcShort( 0xa ), /* Offset= 10 (700) */
/* 716 */ 0x8, /* FC_LONG */
/* 718 */ 0x39, /* FC_ALIGNM8 */
/* 720 */ 0x4c, /* FC_POINTER */
/* 722 */ 0x5b, /* FC_END */
/* 724 */ 0x11, 0x0, /* FC_RP */
/* 726 */ NdrFcShort( 0xfffff10 ), /* Offset= -240 (462) */
/* 728 */ 0x1b, /* FC_CARRAY */
/* 730 */ 0x0, /* 0 */
/* 732 */ NdrFcShort( 0x1 ), /* 1 */
/* 734 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 736 */ 0x0, /*
/* 738 */ NdrFcShort( 0x0 ), /* 0 */
/* 740 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 742 */ 0x6, /* FC_SHORT */
/* 744 */ 0x5b, /* FC_END */
/* 746 */ 0x1a, /*
FC_BOGUS_STRUCT */
/* 748 */ 0x3, /* 3 */
/* 750 */ NdrFcShort( 0x10 ), /* 16 */
/* 752 */ NdrFcShort( 0x0 ), /* 0 */
/* 754 */ NdrFcShort( 0x6 ), /* Offset= 6 (728) */
/* 756 */ 0x8, /* FC_LONG */
/* 758 */ 0x39, /* FC_ALIGNM8 */
/* 760 */ 0x36, /* FC_POINTER */
/* 762 */ 0x5b, /* FC_END */
/* 764 */ 0x12, 0x0, /* FC_UP */
/* 766 */ NdrFcShort( 0xfffffe6 ), /* Offset= -26 (704) */
/* 768 */ 0x1b, /* FC_CARRAY */
/* 770 */ 0x1, /* 1 */
/* 772 */ NdrFcShort( 0x2 ), /* 2 */
/* 774 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 776 */ 0x0, /*
/* 778 */ NdrFcShort( 0x0 ), /* 0 */
/* 780 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 782 */ 0x6, /* FC_SHORT */
/* 784 */ 0x5b, /* FC_END */
/* 786 */ 0x1a, /*
FC_BOGUS_STRUCT */
/* 788 */ 0x3, /* 3 */
/* 790 */ NdrFcShort( 0x10 ), /* 16 */
/* 792 */ NdrFcShort( 0x0 ), /* 0 */
/* 794 */ NdrFcShort( 0x6 ), /* Offset= 6 (756) */
/* 796 */ 0x8, /* FC_LONG */
/* 798 */ 0x39, /* FC_ALIGNM8 */
/* 800 */ 0x36, /* FC_POINTER */
/* 802 */ 0x5b, /* FC_END */
/* 804 */ 0x12, 0x0, /* FC_UP */
/* 806 */ NdrFcShort( 0xfffffe6 ), /* Offset= -26 (732) */
/* 808 */ 0x1b, /* FC_CARRAY */
/* 810 */ 0x3, /* 3 */
/* 812 */ NdrFcShort( 0x4 ), /* 4 */
/* 814 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 816 */ 0x0, /*
/* 818 */ NdrFcShort( 0x0 ), /* 0 */
/* 820 */ NdrFcShort( 0x1 ), /* Corr flags: early, */

```

/* 770 */ 0x8,	/* FC_LONG */	/* 848 */ 0x6,	/* FC_SHORT */
/* 772 */	0x5b,	/* FC_END */	0x6,
FC_BOGUS_STRUCT */	0x1a,	/*	/* FC_SHORT */
/* 774 */ NdrFcShort(0x10),	/* 16 */	/* 850 */ 0x38,	/* FC_ALIGNM4 */
/* 776 */ NdrFcShort(0x0),	/* 0 */	0x8,	/* FC_LONG */
/* 778 */ NdrFcShort(0x6),	/* Offset= 6 (784) */	/* 852 */ 0x8,	/* FC_LONG */
/* 780 */ 0x8,	/* FC_LONG */	0x4c,	/*
/* 782 */ 0x36,	0x39,	FC_EMBEDDED_COMPLEX */	/* 4 */
/* 784 */	/* FC_ALIGNM8 */	/* 854 */ 0x4,	NdrFcShort(0xfffffe0d),
/* 786 */ NdrFcShort(0xfffffe6),	/* FC_POINTER */	-499 (356) */	/* Offset=
/* 788 */	0x5b,	/* FC_END */	0x5b,
/* 790 */ NdrFcShort(0x8),	/* FC_UP */	/* 858 */	0x12, 0x0,
/* 792 */ 0x19,	/* Offset= -26 (760) */	/* 860 */ NdrFcShort(0xfffff02),	/* FC_UP */
/* 794 */ NdrFcShort(0x0),	0x1b,	/* 862 */	0x12, 0x8,
/* 796 */ NdrFcShort(0x1),	/* FC_CARRAY */	/* 864 */ 0x1,	/* FC_UP [simple_pointer] */
/* 798 */ 0xb,	/* 7 */	/* 866 */	/* FC_BYTE */
/* 800 */	/* 8 */	0x5c,	/* FC_PAD */
FC_BOGUS_STRUCT */	/* Corr desc: field pointer, FC_ULONG */	/* 868 */ 0x6,	0x12, 0x8,
/* 802 */ NdrFcShort(0x10),	/* 0 */	/* 870 */	/* FC_UP [simple_pointer] */
/* 804 */ NdrFcShort(0x0),	/* 0 */	/* 872 */ 0x8,	/* FC_LONG */
/* 806 */ NdrFcShort(0x6),	/* Corr flags: early, */	/* 874 */	0x5c,
/* 808 */ 0x8,	/* FC_HYPER */	/* 876 */ 0xa,	/* FC_UP [simple_pointer] */
/* 810 */ 0x36,	0x5b,	/* 878 */	/* FC_FLOAT */
/* 812 */	/* FC_END */	/* 880 */ 0xc,	0x5c,
/* 814 */ NdrFcShort(0xfffffe6),	0x1a,	/* 882 */	/* FC_PAD */
/* 816 */	/*	/* 884 */ NdrFcShort(0xfffffda4),	0x12, 0x0,
/* 818 */ NdrFcShort(0x8),	0x3,	/* 886 */	/* FC_UP */
/* 820 */ 0x8,	/* 3 */	[pointer_deref] */	0x12, 0x10,
/* 822 */ 0x5c,	/* FC_STRUCT */	/* 888 */ NdrFcShort(0xfffffda6),	/* Offset= -602 (286) */
/* 824 */	/* 3 */	/* 890 */	0x12, 0x10,
/* 826 */ NdrFcShort(0x8),	0x15,	[pointer_deref] */	/* FC_UP
/* 828 */ 0x7,	0x3,	/* 892 */ NdrFcShort(0xfffffdbc),	/* Offset= -580 (312) */
/* 830 */ NdrFcShort(0xffc8),	/* 8 */	/* 894 */	0x12, 0x10,
/* 832 */ NdrFcShort(0x1),	/* FC_LONG */	[pointer_deref] */	/* FC_UP
/* 834 */ 0x4c,	0x8,	/* 896 */ NdrFcShort(0xfffffdca),	/* Offset= -566 (330) */
/* 836 */ NdrFcShort(0xfffffec),	/* FC_PAD */	/* 898 */	0x12, 0x10,
/* 838 */ 0x5c,	0x5b,	[pointer_deref] */	/* FC_UP
/* 840 */	/* FC_END */	/* 900 */ NdrFcShort(0xfffffdd8),	/* Offset= -552 (348) */
FC_BOGUS_STRUCT */	0x1a,	/* 902 */	0x12, 0x10,
/* 842 */ NdrFcShort(0x38),	/*	[pointer_deref] */	/* FC_UP
/* 844 */ NdrFcShort(0xfffffec),	0x3,	/* 904 */ NdrFcShort(0x2),	/* Offset= 2 (906) */
/* 846 */ NdrFcShort(0x0),	/* 3 */	/* 906 */	0x12, 0x0,
/* 848 */ 0x6,	/* 56 */	/* 908 */ NdrFcShort(0x16),	/* FC_UP */
/* 850 */ 0x38,	/* Offset= -20 (824) */	/* 910 */	/* Offset= 22 (930) */
/* 852 */ 0x8,	/* Offset= 0 (846) */	0x15,	/* FC_STRUCT */
/* 854 */ 0x4,	/* FC_SHORT */	0x7,	/* 7 */
/* 856 */ 0x6,	/* FC_BYTE */	/* 912 */ NdrFcShort(0x10),	/* 16 */
/* 858 */ 0x8,	/* FC_ALIGNM4 */	/* 914 */ 0x6,	/* FC_SHORT */
/* 860 */ 0x10,	/* FC_BYTE */	0x1,	/* FC_BYTE */
/* 862 */ 0x12,	/* FC_ALIGNM4 */	/* 916 */ 0x1,	/* FC_ALIGNM4 */


```

#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 DEFCLPCKSIZE 4096

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

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

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

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

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

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

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

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

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

/* FUNCTION: int msg_handler(DBPROCESS *dbproc, DBINT msgno, int
msgstate, int severity, char *msgtext)
*/

```

```

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

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

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

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

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

    return 0;
}

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

```

```

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

    return;
}

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

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

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

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

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_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) != SUCCEEDED)
            ThrowError(CDBLIBERR::eDbResults);

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

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

        DiscardNextRows(0);
        DiscardNextResults(0);
    }

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

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

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

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

}

//
}

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

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

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

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

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

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

    throw pDbLibErr;
}

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

    while (TRUE)

```

```

{
    rc = dbnextrow(m_dbproc);
    if (rc == NO_MORE_ROWS)
        break;
    if (rc == FAIL)
    {
        if (iExpectedCount >= 0)

ThrowError(CDBLIBERR::eDbNextRow);
        else
            break;
    }
    iRowsRead++;
}

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

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

    while (TRUE)
    {
        rc = dbresults(m_dbproc);
        if (rc == NO_MORE_RESULTS)
            break;
        if (rc == FAIL)
        {
            if (iExpectedCount >= 0)

ThrowError(CDBLIBERR::eDbResults);
            else
                break;
        }

        DiscardNextRows(-1);
        iResultsRead++;
    }

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

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

    ResetError();

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

```

```

            dbrpcparam(m_dbproc, NULL, 0, SQLINT2,
-1, -1, (BYTE *) &m_txn.StockLevel.w_id); // @w_id smallint
            dbrpcparam(m_dbproc, NULL, 0, SQLINT1,
-1, -1, (BYTE *) &m_txn.StockLevel.d_id); // @d_id tinyint
            dbrpcparam(m_dbproc, NULL, 0, SQLINT2,
-1, -1, (BYTE *) &m_txn.StockLevel.threshold); // @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 &&
sErrTimeoutExpired) != NULL)) &&
                strstr(e->m_msgtext,
                    (++iTryCount <= iMaxRetries))
                {
                    // hit deadlock; backoff for
                    increasingly longer period

                    delete e;
                    Sleep(10 * iTryCount);
                }
                else
                    throw;
            }
        } // while (TRUE)

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

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

    int                iTryCount = 0;
    const BYTE         *pData;

    ResetError();

    while (TRUE)

```

<pre> { try { dbrpcinit(m_dbproc, "tpcc_neworder", 0); dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *) &m_txn.NewOrder.w_id); dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.NewOrder.d_id); dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.NewOrder.c_id); dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.NewOrder.o_ol_cnt); // check whether any order lines are for a remote warehouse m_txn.NewOrder.o_all_local = 1; for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++) { if (m_txn.NewOrder.OL[i].ol_supply_w_id != m_txn.NewOrder.w_id) { m_txn.NewOrder.o_all_local = 0; // at least one remote warehouse break; } } dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.NewOrder.o_all_local); for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++) { dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.NewOrder.OL[i].ol_i_id); dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *) &m_txn.NewOrder.OL[i].ol_supply_w_id); dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *) &m_txn.NewOrder.OL[i].ol_quantity); } if (dbrpcexec(m_dbproc) == FAIL) ThrowError(CDBLIBERR::eDbRpcExec); // Get order line results m_txn.NewOrder.total_amount = 0; for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++) { if (dbresults(m_dbproc) != SUCCEED) ThrowError(CDBLIBERR::eDbResults); if (dbnumcols(m_dbproc) != 5) ThrowError(CDBLIBERR::eWrongNumCols); if (dbnextrow(m_dbproc) != REG_ROW) ThrowError(CDBLIBERR::eDbNextRow); if(pData=dbdata(m_dbproc, 1)) UtilStrCpy(m_txn.NewOrder.OL[i].ol_i_name, pData, dbdatlen(m_dbproc, 1)); if(pData=dbdata(m_dbproc, 2)) </pre>	<pre> 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); </pre>
---	--

<pre> &datetime); = daterec.year; = daterec.month; = daterec.day; = 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 && sErrTimeoutExpired) != NULL)) && { // hit deadlock; backoff for 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; </pre>	<pre> int iTryCount = 0; const BYTE *pData; ResetError(); while (TRUE) { try { dbrpcinit(m_dbproc, "tpcc_payment", 0); dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *) &m_txn.Payment.w_id); dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *) &m_txn.Payment.c_w_id); dbrpcparam(m_dbproc, NULL, 0, SQLFLT8, -1, -1, (BYTE *) &m_txn.Payment.h_amount); dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.Payment.d_id); dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.Payment.c_d_id); dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.Payment.c_id); // if customer id is zero, then payment is by name if (m_txn.Payment.c_id == 0) dbrpcparam(m_dbproc, NULL, 0, SQLCHAR, -1, strlen(m_txn.Payment.c_last), (unsigned char *)m_txn.Payment.c_last); if (dbrpcexec(m_dbproc) == FAIL) ThrowError(CDBLIBERR::eDbRpcExec); if (dbresults(m_dbproc) != SUCCEED) ThrowError(CDBLIBERR::eDbResults); if (dbnextrow(m_dbproc) != REG_ROW) ThrowError(CDBLIBERR::eDbNextRow); if (dbnumcols(m_dbproc) != 27) ThrowError(CDBLIBERR::eWrongNumCols); if (pData=dbdata(m_dbproc, 1)) m_txn.Payment.c_id = *((DBINT *) pData); if (pData=dbdata(m_dbproc, 2)) UtilStrCpy(m_txn.Payment.c_last, pData, dbdatlen(m_dbproc, 2)); if (pData=dbdata(m_dbproc, 3)) { datetime = *((DBDATETIME *) pData); dbdatecrack(m_dbproc, &daterec, &datetime); m_txn.Payment.h_date.year = daterec.year; m_txn.Payment.h_date.month = daterec.month; m_txn.Payment.h_date.day = daterec.day; m_txn.Payment.h_date.hour = daterec.hour; </pre>
--	---

<pre> 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); &datetime; m_txn.Payment.c_since.year = daterec.year; </pre>	<pre> 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 && sErrTimeoutExpired) != NULL)) && (strchr(e->m_msgtext, (++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); } </pre>
--	--

```

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

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

    ResetError();

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

            dbrpcparam(m_dbproc, NULL, 0, SQLINT2,
-1, -1, (BYTE *) &m_txn.OrderStatus.w_id);
            dbrpcparam(m_dbproc, NULL, 0, SQLINT1,
-1, -1, (BYTE *) &m_txn.OrderStatus.d_id);
            dbrpcparam(m_dbproc, NULL, 0, SQLINT4,
-1, -1, (BYTE *) &m_txn.OrderStatus.c_id);

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

            if (dbrpcexec(m_dbproc) == FAIL)

ThrowError(CDBLIBERR::eDbRpcExec);

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

ThrowError(CDBLIBERR::eDbResults);
            }

            if (dbnumcols(m_dbproc) != 5)

ThrowError(CDBLIBERR::eWrongNumCols);

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

ThrowError(CDBLIBERR::eDbNextRow);

                if(pData=dbdata(m_dbproc, 1))

m_txn.OrderStatus.OL[i].ol_supply_w_id = (*(DBSMALLINT *) pData);

```

```

                if(pData=dbdata(m_dbproc, 2))

m_txn.OrderStatus.OL[i].ol_i_id = (*(DBINT *) pData);
                if(pData=dbdata(m_dbproc, 3))

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

m_txn.OrderStatus.OL[i].ol_delivery_d.year = daterec.year;
m_txn.OrderStatus.OL[i].ol_delivery_d.month = daterec.month;
m_txn.OrderStatus.OL[i].ol_delivery_d.day = daterec.day;
m_txn.OrderStatus.OL[i].ol_delivery_d.hour = daterec.hour;
m_txn.OrderStatus.OL[i].ol_delivery_d.minute = daterec.minute;
m_txn.OrderStatus.OL[i].ol_delivery_d.second = daterec.second;
                }
                i++;
                m_txn.OrderStatus.o_ol_cnt = i;

                if (dbresults(m_dbproc) != SUCCEED)

ThrowError(CDBLIBERR::eDbResults);

                if (dbnextrow(m_dbproc) != REG_ROW)

ThrowError(CDBLIBERR::eDbNextRow);

                if (dbnumcols(m_dbproc) != 8)

ThrowError(CDBLIBERR::eWrongNumCols);

                if(pData=dbdata(m_dbproc, 1))
                    m_txn.OrderStatus.c_id =
(*(DBINT *) pData);
                if(pData=dbdata(m_dbproc, 2))

UtilStrCpy(m_txn.OrderStatus.c_last, pData, dbdatlen(m_dbproc,2));
                if(pData=dbdata(m_dbproc, 3))

UtilStrCpy(m_txn.OrderStatus.c_first, pData, dbdatlen(m_dbproc,3));
                if(pData=dbdata(m_dbproc, 4))

UtilStrCpy(m_txn.OrderStatus.c_middle, pData, dbdatlen(m_dbproc, 4));

                if(pData=dbdata(m_dbproc, 5))
                {
                    datetime = *((DBDATETIME *)
pData);
                    dbdatecrack(m_dbproc, &daterec,
&datetime);

                    m_txn.OrderStatus.o_entry_d.year
= daterec.year;

```

```

m_txn.OrderStatus.o_entry_d.month = daterec.month;
m_txn.OrderStatus.o_entry_d.day
= daterec.day;
m_txn.OrderStatus.o_entry_d.hour
= daterec.hour;

m_txn.OrderStatus.o_entry_d.minute = daterec.minute;
m_txn.OrderStatus.o_entry_d.second = daterec.second;
}
if(pData=dbdata(m_dbproc, 6))

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

DiscardNextRows(0);
DiscardNextResults(0);

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

m_txn.OrderStatus.exec_status_code = eOK;

return;
}
catch (CSQLERR *e)
{
if ((e->m_msgno == 1205 ||
(e->m_msgno ==
iErrOleDbProvider &&
sErrTimeoutExpired) != NULL)) &&
{
// hit deadlock; backoff for
increasingly longer period

delete e;
Sleep(10 * iTryCount);
}
else
throw;
}
} // while (TRUE)

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

void CTPCC_DBLIB::Delivery()

```

```

{
int i;
int iTryCount = 0;
const BYTE *pData;

ResetError();

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

dbrpcparam(m_dbproc, NULL, 0, SQLINT2,
-1, -1, (BYTE *) &m_txn.Delivery.w_id);
dbrpcparam(m_dbproc, NULL, 0, SQLINT1,
-1, -1, (BYTE *) &m_txn.Delivery.o_carrier_id);

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

if (dbresults(m_dbproc) != SUCCEEDED)
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 &&
sErrTimeoutExpired) != NULL)) &&
{
// hit deadlock; backoff for
increasingly longer period

delete e;
Sleep(10 * iTryCount);
}
else
throw;
}
} // while (TRUE)

// if (iTryCount)

```

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

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

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

    return;
}
}
```

tpcc_dblib.h

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

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

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

class CSQLERR : public CBaseErr
{
public:

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

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

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

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

class CDBLIBERR : public CBaseErr
{
public:
    enum ACTION
    {
        eNone,
        eUnknown,
        eLogin, //
        eDbOpen, // error from
        eDbUse, //
        eDbSqlExec, //
        eDbSet, //
        eDbNextRow, //
        eWrongRowCount, // more or
        eWrongNumCols, // more or
        eDbResults, //
        eDbRpcExec, //
        eDbSetMaxProcs, // error from
        eDbProcHandler // error from
    };

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

        m_dberrstr = NULL;
        m_oserrstr = NULL;
    };

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

    ACTION m_eAction;
    int m_severity;
    int m_dberror;
    int m_oserr;
    char *m_dberrstr;
};
```

```

char *m_oserrstr;

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

};

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

    CTPCC_DBLIB_ERR( int iErr ) { m_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;};
    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()
{ return &m_txn.StockLevel; };
inline PORDER_STATUS_DATA
BuffAddr_OrderStatus() { return &m_txn.OrderStatus; };

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

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

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

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

```

tpcc_enc.cpp

```

// tpcc_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 tpalloc
    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
*
*/

#ifdef _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.
#ifdef DllDecl
#define DllDecl __declspec( dllimport )
#endif

class CTPCC_ENCINA : public CTPCC_BASE
{
private:
    struct ENC_DATA
    {
        int
        ErrorType;
        int
        error;

        union
        {
            NEW_ORDER_DATA
            Payment_DATA
            DELIVERY_DATA
            STOCK_LEVEL_DATA
            ORDER_STATUS_DATA

        } 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_odbc.cpp

```
/*      FILE:          TPCC_ODBC.CPP
 *
 *      Microsoft TPC-C Kit Ver.
 *
 *      4.20.000
 *
 *      Copyright Microsoft, 1999
 *
 *      All Rights Reserved
 *
 *
 *      Version 4.10.000 audited by
 *      Richard Gimarc, Performance Metrics, 3/17/99
 *
 *      PURPOSE:        Implements ODBC calls for TPC-C txns.
 *      Contact:  Charles Levine (clevine@microsoft.com)
 *
 *      Change history:
 *      4.20.000 - updated rev number to match kit
 *      4.10.001 - not deleting error class in catch handler on
 *      deadlock retry;
 *
 *      not a functional bug, but a
 *      memory leak
 */

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

#define DBNTWIN32
#include <sqltypes.h>
#include <sql.h>
#include <sqltext.h>
#include <odbcss.h>

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

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

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

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

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

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

static SQLHENV henv = SQL_NULL_HENV;
// ODBC environment handle

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

```
        if (
SQLAllocHandleStd(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv)
!= SQL_SUCCESS )

            return FALSE;

        break;

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

            break;

        default:
            /* nothing */;

    }
    return TRUE;
}

/* FUNCTION: CTPCC_ODBC_ERR::ErrorText
 *
 */

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

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

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

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

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

<pre> CTPCC_ODBC::CTPCC_ODBC (LPCSTR szServer, // name of SQL server // LPCSTR szUser, // user name for login // LPCSTR szPassword, // password for login // LPCSTR szHost, // not used // LPCSTR szDatabase // name of database to use //) { RETCODE rc; // initialization m_hdbc = SQL_NULL_HDBC; m_hstmt = SQL_NULL_HSTMT; m_hstmtNewOrder = SQL_NULL_HSTMT; m_hstmtPayment = SQL_NULL_HSTMT; m_hstmtDelivery = SQL_NULL_HSTMT; m_hstmtOrderStatus = SQL_NULL_HSTMT; m_hstmtStockLevel = SQL_NULL_HSTMT; m_descNewOrderCols1 = SQL_NULL_HDESC; m_descNewOrderCols2 = SQL_NULL_HDESC; m_descOrderStatusCols1 = SQL_NULL_HDESC; m_descOrderStatusCols2 = SQL_NULL_HDESC; if (SQLAllocHandle(SQL_HANDLE_DBC, henv, &m_hdbc) != SQL_SUCCESS) ThrowError(CODBCERR::eAllocHandle); if (SQLSetConnectOption(m_hdbc, SQL_PACKET_SIZE, 4096) != SQL_SUCCESS) ThrowError(CODBCERR::eConnOption); { char szConnectStr[256]; char szOutStr[1024]; SQLSMALLINT iOutStrLen; sprintf(szConnectStr, "DRIVER=SQL Server;SERVER=%s;UID=%s;PWD=%s;DATABASE=%s", szServer, szUser, szPassword, szDatabase); rc = SQLDriverConnect(m_hdbc, NULL, (SQLCHAR*)szConnectStr, sizeof(szConnectStr), (SQLCHAR*)szOutStr, sizeof(szOutStr), &iOutStrLen, SQL_DRIVER_NOPROMPT); if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO) ThrowError(CODBCERR::eConnect); } if (SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmt) != SQL_SUCCESS) ThrowError(CODBCERR::eAllocHandle); { char buffer[128]; // set some options affecting connection behavior strcpy(buffer, "set nocount on set XACT_ABORT ON"); </pre>	<pre> rc = SQLExecDirect(m_hstmt, (unsigned char *)buffer, SQL_NTS); if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO) ThrowError(CODBCERR::eExecDirect); // verify that version of stored procs on server is correct char db_sp_version[10]; strcpy(buffer, "{call tpcc_version}"); rc = SQLExecDirect(m_hstmt, (unsigned char *)buffer, SQL_NTS); if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO) ThrowError(CODBCERR::eExecDirect); if (SQLBindCol(m_hstmt, 1, SQL_C_CHAR, &db_sp_version, sizeof(db_sp_version), NULL) != SQL_SUCCESS) ThrowError(CODBCERR::eBindCol); if (SQLFetch(m_hstmt) == SQL_ERROR) ThrowError(CODBCERR::eFetch); if (strcmp(db_sp_version,sVersion)) throw new CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_WRONG_SP_VERSION); SQLFreeHandle(SQL_HANDLE_STMT, m_hstmt); } // Bind parameters for each of the transactions InitNewOrderParams(); InitPaymentParams(); InitOrderStatusParams(); InitDeliveryParams(); InitStockLevelParams(); } CTPCC_ODBC::~CTPCC_ODBC(void) { // note: descriptors are automatically released when the connection is dropped SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtNewOrder); SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtPayment); SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtDelivery); SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtOrderStatus); SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtStockLevel); SQLDisconnect(m_hdbc); SQLFreeHandle(SQL_HANDLE_DBC, m_hdbc); } void CTPCC_ODBC::ThrowError(CODBCERR::ACTION eAction) { RETCODE rc; SDWORD 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) { </pre>
---	--

```

        rc = SQLError(henv, m_hdbc, m_hstmt, (BYTE
*)&szState, &lNativeError,
                                (BYTE *)&szMsg,
sizeof(szMsg), NULL);
        if (rc == SQL_NO_DATA)
            break;

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

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

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

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

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

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

    SQLFreeStmt(m_hstmt, SQL_CLOSE);
    throw pODBCErr;
}

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

    m_hstmt = m_hstmtStockLevel;

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

```

```

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

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

        m_hstmt = m_hstmtStockLevel;

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

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

                SQLFreeStmt(m_hstmt, SQL_CLOSE);

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

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

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

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

        m_hstmt = m_hstmtNewOrder;

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

```

```

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

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

        // 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);
    }

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

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

    m_hstmt = m_hstmtNewOrder;

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

    // clip statement buffer based on number of parameters
    // fixed part is 29 chars and variable part is 6 chars per line item
    i = 29 + m_txn.NewOrder.o_ol_cnt*6;
    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,
(SQLWCHAR*)szSqlTemplate, SQL_NTS);

```

```

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

        // Get order line results
        m_txn.NewOrder.total_amount = 0;
        for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++)
        {
            // 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);
    }

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

        m_hstmt = m_hstmtPayment;

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

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

```

```

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

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

    m_hstmt = m_hstmtPayment;

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

    while (TRUE)
    {
        try

```

```

        {
            rc = SQLExecDirectW(m_hstmt,
(SQLWCHAR*)"L"{call tpcc_payment(?,?,?,?,?,?)}", SQL_NTS);
            if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

                ThrowError(CODBCERR::eExecDirect);

            if ( SQLFetch(m_hstmt) == SQL_ERROR)

                ThrowError(CODBCERR::eFetch);

            SQLFreeStmt(m_hstmt, SQL_CLOSE);

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

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

                throw;

            // hit deadlock; backoff for increasingly longer
period

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

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

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

    m_hstmt = m_hstmtOrderStatus;

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

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SSHORT, SQL_SMALLINT, 0, 0, &m_txn.OrderStatus.w_id, 0,
NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.OrderStatus.d_id, 0, NULL) != SQL_SUCCESS

```

```

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

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

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

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

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

void CTPCC_ODBC::OrderStatus()
{

```

```

    int
    iTryCount = 0;
    RETCODE

rc;

    m_hstmt = m_hstmtOrderStatus;

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

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

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

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

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

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

            m_txn.OrderStatus.o_ol_cnt =
(short)m_RowsFetched;

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

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

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

```

```

        SQLFreeStmt(m_hstmt, SQL_CLOSE);

        if (m_txn.OrderStatus.o_ol_cnt == 0)
            throw new CTPCC_ODBC_ERR(
CTPCC_ODBC_ERR::ERR_NO_SUCH_ORDER );
        else if (m_txn.OrderStatus.c_id == 0 &&
m_txn.OrderStatus.c_last[0] == 0)
            throw new CTPCC_ODBC_ERR(
CTPCC_ODBC_ERR::ERR_INVALID_CUST );
        else
            break;
    }
    catch (CODBCERR *e)
    {
        if ((!e->m_bDeadLock) || (++iTryCount >
iMaxRetries))
            throw;

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

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

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

    m_hstmt = m_hstmtDelivery;

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

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

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

    m_hstmt = m_hstmtDelivery;

    while (TRUE)

```

```

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

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

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

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

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

tpcc_odbc.h

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

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

class CODBCERR : public CBaseErr
{
public:
    enum ACTION
    {

```

	eNone,			ERR_NO_SUCH_ORDER,	
	eUnknown,		// "No orders found for customer."		
	eAllocConn,	//		ERR_RETRIED_TRANS,	
error from SQLAllocConnect				// "Retries before transaction succeeded."	
	eAllocHandle,	// error from		};	
SQLAllocHandle					
	eConnOption,	// error from		CTPCC_ODBC_ERR(int iErr) { m_erno = iErr;	
SQLSetConnectOption			m_iTryCount = 0; };		
	eConnect,	// error from			
SQLConnect				CTPCC_ODBC_ERR(int iErr, int iTryCount) {	
error from SQLAllocStmt	eAllocStmt,	//	m_erno = iErr; m_iTryCount = iTryCount; };		
	eExecDirect,	// error from		int m_erno;	
SQLExecDirect				int m_iTryCount;	
	eBindParam,	//			
error from SQLBindParameter				int ErrorType() {return ERR_TYPE_TPCC_ODBC;};	
	eBindCol,	// error from		int ErrorNum() {return m_erno;};	
SQLBindCol					
	eFetch,	//		char *ErrorText();	
error from SQLFetch			};		
	eFetchScroll,	// error from			
SQLFetchScroll				class DllDecl CTPCC_ODBC : public CTPCC_BASE	
	eMoreResults,	// error from		{	
SQLMoreResults				private:	
	ePrepare,	// error from		// declare variables and private functions here...	
SQLPrepare				BOOL m_bDeadlock;	//
	eExecute,	// error from	transaction was selected as deadlock victim		
SQLExecute			int m_MaxRetries;		
	eSetEnvAttr,	// error from	// retry count on deadlock		
SQLSetEnvAttr					
	eSetStmtAttr	// error from		SQLHENV m_henv;	
SQLSetStmtAttr				// ODBC environment handle	
				SQLHDBC m_hdbc;	
				SQLHSTMT m_hstmt;	//
			the current hstmt		
				SQLHSTMT m_hstmtNewOrder;	
				SQLHSTMT m_hstmtPayment;	
				SQLHSTMT m_hstmtDelivery;	
				SQLHSTMT m_hstmtOrderStatus;	
				SQLHSTMT m_hstmtStockLevel;	
				SQLHDESC m_descNewOrderCols1;	
				SQLHDESC m_descNewOrderCols2;	
				SQLHDESC m_descOrderStatusCols1;	
				SQLHDESC m_descOrderStatusCols2;	
				// new-order specific fields	
				SQLINTEGER m_BindOffset;	
				SQLINTEGER m_RowsFetched;	
				int m_no_commit_flag;	
				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;


```

};

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."
    };

```

```

    }
    m_txn;

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

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

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

};

```

```

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

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

```

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 definitions 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 WINAPIENTRY 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 = tperrno;
                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 = tperrno;
        LeaveCriticalSection(&TpCriticalSection);

        if (iRc < 0)
            throw new CTUXERR( tperrno );

        int value = 1;
        TlsSetValue(TLSIsTpInitKey,&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( tperrno );
}

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( tperrno );

    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( tperrno );

    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 = tpacall("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
        int
        union
        {
            NEW_ORDER_DATA
            PAYMENT_DATA
            DELIVERY_DATA
        }
    };
    ErrorType;
    error;
    NewOrder;
    Payment;
    Delivery;

```

```

        StockLevel;
        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_erno = iErr;
        m_iErrorType = 0;
        m_iError = GetLastError(); // only
meaningful if m_erno == TPEOS
    };

    // use this interface to impersonate a non-Tuxedo error
type
    CTUXERR( int iErrorType, int iError )
    {
        m_iErrorType = iErrorType;
        m_iError = iError;
        m_erno = 0;
    }

    int m_erno;
    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)();

```

trans.h

```

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

// String length constants
#define SERVER_NAME_LEN      20
#define DATABASE_NAME_LEN   20
#define USER_NAME_LEN       20
#define PASSWORD_LEN        20
#define TABLE_NAME_LEN    20
#define I_DATA_LEN          50
#define I_NAME_LEN          24
#define BRAND_LEN           1
#define LAST_NAME_LEN       16
#define W_NAME_LEN          10
#define ADDRESS_LEN         20
#define STATE_LEN           2
#define ZIP_LEN             9
#define S_DIST_LEN          24
#define S_DATA_LEN          50
#define D_NAME_LEN          10
#define FIRST_NAME_LEN      16
#define MIDDLE_NAME_LEN     2
#define PHONE_LEN           16
#define DATETIME_LEN        30
#define CREDIT_LEN          2
#define C_DATA_LEN          250
#define H_DATA_LEN          24
#define DIST_INFO_LEN       24
#define MAX_OL_NEW_ORDER_ITEMS 15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN          25
#define OL_DIST_INFO_LEN    24

// TIMESTAMP_STRUCT is provided by the ODBC header file sqltypes.h, but
is not available

```

```

// when compiling with dblib, so redefined here. Note: we are using the symbol
"__SQLTYPES"
// (declared in sqltypes.h) as a way to determine if TIMESTAMP_STRUCT has
been declared.
#ifndef __SQLTYPES
typedef struct
{
    short
    /*
    SQLSMALLINT */ year;
    unsigned short
    /*
    month;
    unsigned short
    /*
    day;
    unsigned short
    /*
    hour;
    unsigned short
    /*
    minute;
    unsigned short
    /*
    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
    short ol_supply_w_id;
    long ol_i_id;
    short ol_quantity;

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

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

    // output params
    EXEC_STATUS exec_status_code;
    char c_last[ LAST_NAME_LEN+1 ];
    char c_credit[ CREDIT_LEN+1 ];
    double c_discount;
    double w_tax;
    double d_tax;
    long o_id;
    short o_commit_flag;
}

```

```

        TIMESTAMP_STRUCT      o_entry_d;
        short                  o_all_local;
        double                  total_amount;
        OL_NEW_ORDER_DATA
    OL[MAX_OL_NEW_ORDER_ITEMS];
} NEW_ORDER_DATA, *PNEW_ORDER_DATA;

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

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

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

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

    // output params
    EXEC_STATUS    exec_status_code;
    char           c_first[FIRST_NAME_LEN+1];
    char           c_middle[MIDDLE_NAME_LEN+1];
    double         c_balance;

```

```

        long           o_id;
        TIMESTAMP_STRUCT      o_entry_d;
        short          o_carrier_id;
        OL_ORDER_STATUS_DATA
    OL[MAX_OL_ORDER_STATUS_ITEMS];
        short          o_ol_cnt;
} ORDER_STATUS_DATA, *PORDER_STATUS_DATA;

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

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

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

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

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

```

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 definitions 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
 * -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
    int
    union
    {
        NEW_ORDER_DATA
        PAYMENT_DATA
        DELIVERY_DATA
        STOCK_LEVEL_DATA
        ORDER_STATUS_DATA
    } 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 *));
#ifdef __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 tnull_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 = &tnull_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()));
}

```

txn_base.h

```

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

#pragma once

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

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

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

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

txnlog.h

/* FILE: TXNLOG.H
 * Microsoft TPC-C Kit Ver.
4.10.000
 *
 * NOTE: this file is RTE specific
and should not be included
 * in Full Disclosure Reports.
 *
 * Copyright Microsoft, 1999
 *
 * PURPOSE: Structure definitions for logging delivery txn
completion stats.
 * Contact: Charles Levine (clevine@microsoft.com)
 */

typedef struct _TXN_NEWORDER

```

```

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

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

typedef struct _TXN_ORDERSTATUS
{
    BYTE      CustByName;
} TXN_ORDERSTATUS;

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

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

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

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

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

// TPC-C Txn Record Layout:
//
//TxnStartT0' is a Julian timestamp corresponding to the moment the
//txn is sent to the SUT, i.e., beginning of response time. Deltas
//are in milliseconds. Note that if RTDelay > 0, then the txn was
//delayed by this amount. The delay occurs at the beginning of the

//response time. So if RTDelay > 0, then the txn was actually sent
//at TxnStartT0 + RTDelay.
//
//Graphically:
//
// time -->
//
// |--- Menu ---| Keying --| Response --| Think --|
// <- DeltaT1 -> <- DeltaT2 -> <- DeltaT4 -> <- DeltaT3 ->
// ^
// ^ TxnStartT0
//
//RTDelay is the amount of response time delay included in DeltaT4.
//RTDelay is recorded per txn because this value can be changed on
//the fly, and so may vary from txn to txn.
//
//TxnStatus is the txn completion code. It is used to indicate errors.
//For example, in the New Order txn, 1% of txns abort. TxnStatus will
//reflect this.

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

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

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

```

```

= 0          BYTE    TxnSubType;                // #define    MAX_NUM_BUFFERS                2

// end of common header

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

time (ms)    WORD    w_id;                // warehouse

ID          BYTE    TxnStatus;          // completion status for

txn to indicate errors    BYTE    reserved;          // for word alignment
short    o_carrier_id;    // carrier id
long    o_id[10];          // returned delivery

transaction ids
    } TXN_RECORD_TPCC_DELIV_DEF,
*PTXN_RECORD_TPCC_DELIV_DEF;

#define    TXN_LOG_VERSION                1
#define    TXN_DATA_START                4096    // offset in
log file where log records start
#define    TXN_LOG_EYE_CATCHER    "BC"    // signature
bytes at the start of log file

//////////
// The transaction log has a header as the first 4K block.
//
typedef struct _TXN_LOG_HEADER
{
    char                EyeCatcher[2];    //
signature bytes; should always be "BC"
    int                LogVersion;
// set to TXN_LOG_VERSION
    JULIAN_TIME        BeginTxnTS;
// timestamp of first (lowest) txn start
    JULIAN_TIME        EndTxnTS;
// timestamp of last (highest) txn completion time
    int                iRecCount;
// number of records in log file
    BOOL                bLogSorted;
    int                iFileSize;
// file size in bytes

// the record map provides a fast way to get close to a
particular timestamp in a sorted log file.
//
//
//
// timestamp of record
    JULIAN_TIME        TS;
//
    int
iPos;                // byte position in file
//
}
RecMap[RecMapSize];
// #define    RecMapSize                200

} TXN_LOG_HEADER, *PTXN_LOG_HEADER;

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

#define    NUM_READ_BUFFERS                1
#define    NUM_WRITE_BUFFERS                2

// flags passed in to the constructor
#define    TXN_LOG_WRITE                0x01
#define    TXN_LOG_READ                0x02
#define    TXN_LOG_SORTED                0x04

#define    TXN_LOG_OS_ERROR                1
#define    TXN_LOG_NOT_SORTED                2

#define    SKIP_CTRL_RECS                1

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

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

    JULIAN_TIME        lastTS;
//when writing sorted output, used to verify records are sorted
    BOOL                bWrite;
//writing log file

    BOOL                bLogSorted;
// is log file sorted? applies to both input and output
    JULIAN_TIME        BeginTxnTS;
// timestamp of first (lowest) txn start
    JULIAN_TIME        EndTxnTS;
// timestamp of last (highest) txn completion time
    int                iRecCount;
// number of records in log file

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

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

    DWORD                dwError;
    HANDLE                hTxnFile;
//handle to log file

```

```

        HANDLE          hMapFile;
//map file used when sorting the log
        HANDLE          hIoComplete;
//event to signify that there are no pending IOs
        HANDLE          hLogFileIo;
//event to signal the IO thread to write the inactive buffer

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

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

public:
        CTxnLog::CTxnLog(LPCTSTR szFileName, DWORD
dwOpts);

        ~CTxnLog(void);

        int WriteToLog(PTXN_RECORD_TPCC pTxnRcrd);
        int WriteToLog(PTXN_RECORD_TPCC_DELIV_DEF
pTxnRcrd);

        int WriteToLog(PTXN_RECORD_CONTROL
pCtrlRec);

        int WriteToLog(PTXN_RECORD_HEADER pCtrlRec);

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

        void CloseTransactionLogFile(void);

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

        int Sort(void);
        PTXN_RECORD_HEADER GetSortedRecord(int
index);

        inline BOOL IsSorted(void) { return bLogSorted; };
        inline JULIAN_TIME BeginTS(void) { return

BeginTxnTS; };

        inline JULIAN_TIME EndTS(void) { return EndTxnTS;

};

        inline int RecordCount(void) { return iRecCount; };

class CTXNLOG_ERR : public CBaseErr
{
public:
        enum CTPCC_DBLIB_ERRS
        {
                ERR_BAD_FILE_FORMAT = 1,
                ERR_UNKNOWN_LOG_VERSION,
                ERR_BROKEN_LOG_FILE,
                ERR_LOG_NOT_SORTED,
                ERR_INVALID_TIME_SEQ,
        };

        CTXNLOG_ERR( int iErr ) { m_errno = iErr; };

        int          m_errno;

```

```

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

// TODO: need to complete...
char *ErrorText() {return "";};
};

```

webclnt.dsp

```

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

# TARGETTYPE "Win32 (x86) Application" 0x0101

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

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

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

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

```

```
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib /nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib /nologo /subsystem:windows /machine:I386

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

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

!ENDIF

# Begin Target

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

weclnt.dsw

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

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

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

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

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

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

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

#####
#####

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

Package=<5>
{{{
}}}

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

#####
#####

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

Package=<5>
{{{
}}}

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

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

<pre> Begin Project Dependency Project_Dep_Name tm_com_dll End Project Dependency Begin Project Dependency Project_Dep_Name tm_encina_dll End Project Dependency }}} ##### ##### Project: "tm_com_dll"=. \tm_com_dll\tm_com_dll.dsp - Package Owner=<4> Package=<5> {{{ }}} Package=<4> {{{ Begin Project Dependency Project_Dep_Name tpcc_com_ps End Project Dependency Begin Project Dependency Project_Dep_Name tpcc_com_all End Project Dependency }}} ##### ##### Project: "tm_encina_dll"=. \tm_encina_dll\tm_encina_dll.dsp - Package Owner=<4> Package=<5> {{{ }}} Package=<4> {{{ }}} ##### ##### Project: "tm_tuxedo_dll"=. \tm_tuxedo_dll\tm_tuxedo_dll.dsp - Package Owner=<4> Package=<5> {{{ }}} Package=<4> {{{ }}} ##### ##### Project: "tpcc_com_all"=. \tpcc_com_all\tpcc_com_all.dsp - Package Owner=<4> Package=<5> {{{ }}} Package=<4> {{{ Begin Project Dependency </pre>	<pre> Project_Dep_Name tpcc_com_ps End Project Dependency }}} ##### ##### Project: "tpcc_com_ps"=. \tpcc_com_ps\tpcc_com_ps.dsp - Package Owner=<4> Package=<5> {{{ }}} Package=<4> {{{ }}} ##### ##### Project: "tuxapp"=. \tuxapp\tuxapp.dsp - Package Owner=<4> Package=<5> {{{ }}} Package=<4> {{{ Begin Project Dependency Project_Dep_Name db_dblib_dll End Project Dependency Begin Project Dependency Project_Dep_Name db_odbc_dll End Project Dependency }}} ##### ##### Global: Package=<5> {{{ }}} Package=<3> {{{ }}} ##### ##### Stored Procedures <i>neword.sql</i> -- File: NEWORD.SQL -- Microsoft TPC-C Benchmark Kit Ver. 4.22 -- Copyright Microsoft, 2001 -- Purpose: Creates new order transaction stored procedure -- -- Interface Level: 4.10.000 use tpcc go </pre>
---	---

```

if exists ( select name from sysobjects where name = "tpcc_neworder" )
    drop procedure tpcc_neworder
go

```

```

create proc tpcc_neworder

```

```

    @w_id      smallint,
    @d_id      tinyint,
    @c_id      int,
    @o_ol_cnt  tinyint,
    @o_all_local tinyint,
    @i_id1     int = 0,

    @i_id2     int = 0,

    @i_id3     int = 0,

    @i_id4     int = 0,

    @i_id5     int = 0,

    @i_id6     int = 0,

    @i_id7     int = 0,

    @i_id8     int = 0,

    @i_id9     int = 0,

    @i_id10    int = 0,

    @i_id11    int = 0,

    @i_id12    int = 0,

    @i_id13    int = 0,

    @i_id14    int = 0,

    @i_id15    int = 0,

    @s_w_id1   smallint = 0, @ol_qty1  smallint = 0,

    @s_w_id2   smallint = 0, @ol_qty2  smallint = 0,

    @s_w_id3   smallint = 0, @ol_qty3  smallint = 0,

    @s_w_id4   smallint = 0, @ol_qty4  smallint = 0,

    @s_w_id5   smallint = 0, @ol_qty5  smallint = 0,

    @s_w_id6   smallint = 0, @ol_qty6  smallint = 0,

    @s_w_id7   smallint = 0, @ol_qty7  smallint = 0,

    @s_w_id8   smallint = 0, @ol_qty8  smallint = 0,

    @s_w_id9   smallint = 0, @ol_qty9  smallint = 0,

    @s_w_id10  smallint = 0, @ol_qty10 smallint = 0,

    @s_w_id11  smallint = 0, @ol_qty11 smallint = 0,

    @s_w_id12  smallint = 0, @ol_qty12 smallint = 0,

    @s_w_id13  smallint = 0, @ol_qty13 smallint = 0,

    @s_w_id14  smallint = 0, @ol_qty14 smallint = 0,

    @s_w_id15  smallint = 0, @ol_qty15 smallint = 0

```

```

as
declare  @w_tax      numeric(4,4),
    @d_tax      numeric(4,4),
    @c_last     char(16),
    @c_credit   char(2),
    @c_discount numeric(4,4),
    @i_price     numeric(5,2),
    @i_name      char(24),
    @i_data     char(50),
    @o_entry_d   datetime,
    @remote_flag int,
    @s_quantity  smallint,
    @s_data      char(50),
    @s_dist      char(24),
    @li_no       int,
    @o_id        int,
    @commit_flag tinyint,
    @li_id       int,
    @li_s_w_id   smallint,
    @li_qty       smallint,
    @ol_number   int,
    @c_id_local  int

```

```

begin

```

```

begin transaction n

```

```

-- get district tax and next 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 (tablock 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,

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

@li_s_w_id,
1899",
"dec 31,
@li_qty,
@i_price
@s_dist)

* @li_qty,

-- 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

-- get customer last name, discount, and credit rating

select    @c_last = c_last,
          @c_discount = c_discount,
          @c_credit = c_credit,
          @c_id_local = c_id
from      customer (repeatableread)
where     c_id = @c_id and
          c_w_id = @w_id and
          c_d_id = @d_id

-- insert fresh row into orders table

insert into orders values (    @o_id,
                              @d_id,
                              @w_id,
                              @c_id_local,
                              @o_entry_d,
                              0,
                              @o_ol_cnt,
                              @o_all_local)

-- insert corresponding row into new-order table

insert into new_order values (    @o_id,
                              @d_id,
                              @w_id)

-- select warehouse tax

```

```

select      @w_tax   = w_tax
from        warehouse (repeatableread)
where       w_id     = @w_id

if (@commit_flag = 1)

        commit transaction n
else

-- all that work for nuthn!!!

        rollback transaction n

-- return order data to client

select      @w_tax,
            @d_tax,
            @o_id,
            @c_last,
            @c_discount,
            @c_credit,
            @o_entry_d,
            @commit_flag

end

go

```

payment.sql

```

-- File:   PAYMENT.SQL
--         Microsoft TPC-C Benchmark Kit Ver. 4.22
--         Copyright Microsoft, 2001
-- Purpose: Creates payment transaction stored procedure
--
--         Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_payment" )
        drop procedure tpcc_payment
go

create proc tpcc_payment      @w_id      smallint,
                             @c_w_id    smallint,
                             @h_amount  numeric(6,2),
                             @d_id      tinyint,
                             @c_d_id    tinyint,
                             @c_id      int,
                             @c_last    char(16) = ""

as
declare @w_street_1 char(20),
        @w_street_2 char(20),
        @w_city     char(20),
        @w_state     char(2),
        @w_zip       char(9),
        @w_name      char(10),
        @d_street_1 char(20),
        @d_street_2 char(20),
        @d_city      char(20),
        @d_state     char(2),
        @d_zip       char(9),
        @d_name      char(10),

```

```

@c_first   char(16),
@c_middle  char(2),
@c_street_1 char(20),
@c_street_2 char(20),
@c_city    char(20),
@c_state   char(2),
@c_zip     char(9),
@c_phone   char(16),
@c_since   datetime,
@c_credit  char(2),
@c_credit_lim numeric(12,2),
@c_balance numeric(12,2),
@c_discount numeric(4,4),
@data      char(500),
@c_data    char(500),
@datetime  datetime,
@w_ytd     numeric(12,2),
@d_ytd     numeric(12,2),
@cnt       smallint,
@val       smallint,
@screen_data char(200),
        @d_id_local tinyint,
        @w_id_local smallint,
        @c_id_local int

```

```
select @screen_data = ""
```

```
begin tran p
```

```
-- get payment date
```

```
select      @datetime = getdate()
```

```
if (@c_id = 0)
begin
```

```
-- get customer id and info using last name
```

```
select      @cnt      = count(*)
from        customer (repeatableread)
where       c_last    = @c_last and
            c_w_id    = @c_w_id and
            c_d_id    = @c_d_id

```

```
select      @val = (@cnt + 1) / 2
set         rowcount @val

```

```
select      @c_id     = c_id
from        customer (repeatableread)
where       c_last    = @c_last and
            c_w_id    = @c_w_id and
            c_d_id    = @c_d_id
order       by c_last, c_first

```

```
set         rowcount 0

```

```
end
```

```
-- get customer info and update balances
```

```

update      customer
set         @c_balance      = c_balance      = c_balance
- @h_amount,
            c_payment_cnt   = c_payment_cnt + 1,
            c_ytd_payment   = c_ytd_payment + @h_amount,
            @c_first        = c_first,
            @c_middle       = c_middle,
            @c_last         = c_last,

```

```

        @c_street_1      = c_street_1,
        @c_street_2      = c_street_2,
        @c_city           = c_city,
        @c_state          = c_state,
        @c_zip            = c_zip,
        @c_phone          = c_phone,
        @c_credit          = c_credit,
        @c_credit_lim     = c_credit_lim,
        @c_discount       = c_discount,
        @c_since          = c_since,
        @data             = c_data,
        @c_id_local       = c_id
    where
        c_id              = @c_id and
        c_w_id             = @c_w_id and
        c_d_id            = @c_d_id

-- if customer has bad credit get some more info

    if (@c_credit = "BC")
    begin

--        compute new info

        select @c_data      = convert(char(5),@c_id) +
                                convert(char(4),@c_d_id) +
                                convert(char(5),@c_w_id) +
                                convert(char(4),@d_id) +
                                convert(char(5),@w_id) +
                                convert(char(19),@h_amount) +
                                substring(@data, 1, 458)

-- update customer info

        update      customer
        set         c_data      = @c_data

        where       c_id        = @c_id and
                    c_w_id      = @c_w_id and
                    c_d_id      = @c_d_id

        select      @screen_data = substring (@c_data,1,200)

    end

-- get district data and update year-to-date

    update      district
    set         d_ytd          = d_ytd + @h_amount,
                @d_street_1    = d_street_1,
                @d_street_2    = d_street_2,
                @d_city         = d_city,
                @d_state        = d_state,
                @d_zip          = d_zip,
                @d_name         = d_name,
                @d_id_local     = d_id
    where       d_w_id         = @w_id and
                d_id           = @d_id

-- get warehouse data and update year-to-date

    update      warehouse
    set         w_ytd          = w_ytd + @h_amount,
                @w_street_1    = w_street_1,
                @w_street_2    = w_street_2,
                @w_city         = w_city,
                @w_state        = w_state,
                @w_zip          = w_zip,

        @w_name              = w_name,
        @w_id_local          = w_id
    where
        w_id                 = @w_id

-- create history record

        insert into history values (
                                @c_id_local,
                                @c_d_id,
                                @c_w_id,
                                @d_id_local,
                                @w_id_local,
                                @datetime,
                                @h_amount,
                                @w_name + " " +
                                @d_name)
    commit tran p

-- return data to client

select      @c_id,
            @c_last,
            @datetime,
            @w_street_1,
            @w_street_2,
            @w_city,
            @w_state,
            @w_zip,
            @d_street_1,
            @d_street_2,
            @d_city,
            @d_state,
            @d_zip,
            @c_first,
            @c_middle,
            @c_street_1,
            @c_street_2,
            @c_city,
            @c_state,
            @c_zip,
            @c_phone,
            @c_since,
            @c_credit,
            @c_credit_lim,
            @c_discount,
            @c_balance,
            @screen_data

go

ordstat.sql

-- File:   ORDSTAT.SQL
--         Microsoft TPC-C Benchmark Kit Ver. 4.22
--         Copyright Microsoft, 2001
-- Purpose: Creates order status transaction stored procedure
--
--         Interface Level: 4.10.000

use tpcc
go

if exists ( select name from sysobjects where name = "tpcc_orderstatus" )
    drop procedure tpcc_orderstatus
go

create proc tpcc_orderstatus    @w_id    smallint,

```

```

        @d_id    tinyint,

        @c_id    int,
        @c_last  char(16) = ""

as

declare @c_balance    numeric(12,2),
        @c_first    char(16),
        @c_middle    char(2),
        @o_id        int,
        @o_entry_d    datetime,
        @o_carrier_id smallint,
        @cnt          smallint

begin tran o

if (@c_id = 0)
    begin

-- get customer id and info using last name

        select    @cnt    = (count(*)+1)/2
        from      customer (repeatableread)
        where     c_last  = @c_last and
                  c_w_id  = @w_id and
                  c_d_id  = @d_id

        set       rowcount @cnt

        select    @c_id    = c_id,
                  @c_balance = c_balance,
                  @c_first = c_first,
                  @c_last  = c_last,
                  @c_middle = c_middle
        from      customer (repeatableread)
        where     c_last  = @c_last and
                  c_w_id  = @w_id and
                  c_d_id  = @d_id

        order    by c_w_id, c_d_id, c_last, c_first

        set       rowcount 0

    end

    else

        begin

-- get customer info if by id

        select    @c_balance    = c_balance,
                  @c_first    = c_first,
                  @c_middle    = c_middle,
                  @c_last     = c_last
        from      customer (repeatableread)
        where     c_id        = @c_id and
                  c_d_id      = @d_id and
                  c_w_id      = @w_id

        select    @cnt    = @@rowcount

    end

-- if no such customer

    if (@cnt = 0)
        begin
            raiserror("Customer not found",18,1)
        end
    end

```

```

        goto custnotfound
    end

-- get order info

        select    @o_id        = o_id,
                  @o_entry_d    = o_entry_d,
                  @o_carrier_id = o_carrier_id
        from      orders (serializable)
        where     o_c_id        = @c_id and
                  o_d_id        = @d_id and
                  o_w_id        = @w_id

        order    by o_id asc

-- select order lines for the current order

        select    ol_supply_w_id,
                  ol_i_id,
                  ol_quantity,
                  ol_amount,
                  ol_delivery_d
        from      order_line (repeatableread)
        where     ol_o_id = @o_id and
                  ol_d_id = @d_id and
                  ol_w_id = @w_id

custnotfound:

commit tran o

-- return data to client

select    @c_id,
        @c_last,
        @c_first,
        @c_middle,
        @o_entry_d,
        @o_carrier_id,
        @c_balance,
        @o_id

go

```

delivery.sql

```

-- File:  DELIVERY.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.22
-- Copyright Microsoft, 2001
-- Purpose:  Creates delivery transaction stored procedure
--
-- Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_delivery" )
    drop procedure tpcc_delivery
go

create proc tpcc_delivery    @w_id    smallint,

                                @o_carrier_id smallint

as

declare @d_id    tinyint,
        @o_id    int,
        @c_id    int,

```

```

@total      numeric(12,2),
@oid1       int,
@oid2       int,
@oid3       int,
@oid4       int,
@oid5       int,
@oid6       int,
@oid7       int,
@oid8       int,
@oid9       int,
@oid10      int

select @d_id = 0

begin tran d

    while (@d_id < 10)
    begin

        select @d_id = @d_id + 1,
               @total = 0,
               @o_id = 0

        select top 1
               @o_id = no_o_id
        from    new_order (serializable 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 tran d

    -- return delivery data to client

    select @oid1,
           @oid2,
           @oid3,
           @oid4,
           @oid5,
           @oid6,
           @oid7,
           @oid8,
           @oid9,
           @oid10

go

```

stocklev.sql

```

-- File: STOCKLEV.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.22
-- Copyright Microsoft, 2001
-- Purpose: Creates stock level transaction stored procedure
--
-- Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_stocklevel" )
    drop procedure tpcc_stocklevel
go

create proc tpcc_stocklevel @w_id smallint,
                           @d_id tinyint,
                           @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

go

```

version.sql

```

-- File:   VERSION.SQL
--         Microsoft TPC-C Benchmark Kit Ver. 4.22
--         Copyright Microsoft, 2001
-- Purpose: Returns version level of TPC-C stored procs
-- Note:   Always update the return value of this proc for
--         any interface changes or "must have" bug fixes.
--
-- The value returned by this SP defines the "interface level",
-- which must match between the stored procs and the client code.
-- The interface level may be down rev from the current kit. This
-- indicates that the interface hasn't changed since that version.

use tpcc
go

if exists ( select name from sysobjects where name = "tpcc_version" )
    drop procedure tpcc_version
go

create proc tpcc_version
as
declare    @version char(8)

begin
    select @version = "4.10.000"
    select @version as "Version"
end

go

```

null-txn.sql

```

-- TPC-C Null Txn Stored Procs
-- Microsoft TPC-C Kit
-- 8/17/99
--
-- 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 advan-
-- tage is that these
-- stored procs place almost no load on SQL Server and do not require a
-- database.
--
-- The purpose of these stored procs is to size and test the web client without the
-- need of a fully
-- scaled database.
--

```

```

drop proc tpcc_delivery
drop proc tpcc_neworder
drop proc tpcc_orderstatus
drop proc tpcc_payment
drop proc tpcc_stocklevel
drop proc tpcc_version
drop table order_line_null
go

```

```

create proc tpcc_delivery    @w_id    smallint,

                             @o_carrier_id smallint
as

```

```

declare @d_id tinyint,
        @o_id int,
        @c_id int,
        @total numeric(12,2),
        @oid1 int,
        @oid2 int,
        @oid3 int,
        @oid4 int,
        @oid5 int,
        @oid6 int,
        @oid7 int,
        @oid8 int,
        @oid9 int,
        @oid10 int

```

```

declare @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 proc tpcc_neworder

```

```

                             @w_id    smallint,
                             @d_id    tinyint,
                             @c_id    int,
                             @o_ol_cnt tinyint,
                             @o_all_local tinyint,
                             @i_id1 int = 0,

                             @i_id2 int = 0,
                             @i_id3 int = 0,
                             @i_id4 int = 0,
                             @i_id5 int = 0,
                             @i_id6 int = 0,
                             @i_id7 int = 0,
                             @i_id8 int = 0,
                             @i_id9 int = 0,
                             @i_id10 int = 0,

                             @s_w_id1 smallint = 0, @ol_qty1 smallint = 0,
                             @s_w_id2 smallint = 0, @ol_qty2 smallint = 0,
                             @s_w_id3 smallint = 0, @ol_qty3 smallint = 0,
                             @s_w_id4 smallint = 0, @ol_qty4 smallint = 0,
                             @s_w_id5 smallint = 0, @ol_qty5 smallint = 0,
                             @s_w_id6 smallint = 0, @ol_qty6 smallint = 0,
                             @s_w_id7 smallint = 0, @ol_qty7 smallint = 0,
                             @s_w_id8 smallint = 0, @ol_qty8 smallint = 0,
                             @s_w_id9 smallint = 0, @ol_qty9 smallint = 0,
                             @s_w_id10 smallint = 0, @ol_qty10 smallint = 0,

```

```

@s_w_id11 smallint = 0, @ol_qty11 smallint = 0,
@s_w_id12 smallint = 0, @ol_qty12 smallint = 0,
@s_w_id13 smallint = 0, @ol_qty13 smallint = 0,
@s_w_id14 smallint = 0, @ol_qty14 smallint = 0,
@s_w_id15 smallint = 0, @ol_qty15 smallint = 0

as
declare @w_tax numeric(4,4),
        @d_tax numeric(4,4),
        @c_last char(16),
        @c_credit char(2),
        @c_discount numeric(4,4),
        @i_price numeric(5,2),
        @i_name char(24),
        @o_entry_d datetime,
        @li_no int,
        @o_id int,
        @commit_flag tinyint,
        @li_id int,
        @li_qty smallint

declare @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 proc tpcc_orderstatus @w_id smallint,
@o_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

declare @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 proc tpcc_payment @w_id      smallint,
```

```
    smallint,
```

```
    numeric(6,2),
```

```
    tinyint,
```

```
    tinyint,
```

```
    int,
```

```
    char(16) = "
```

as

```
declare @w_street_1 char(20),
```

```
    @w_street_2 char(20),
```

```
    @w_city char(20),
```

```
    @w_state char(2),
```

```
    @w_zip char(9),
```

```
    @w_name char(10),
```

```
    @d_street_1 char(20),
```

```
    @d_street_2 char(20),
```

```
    @d_city char(20),
```

```
    @d_state char(2),
```

```
    @d_zip char(9),
```

```
    @d_name char(10),
```

```
    @c_first char(16),
```

```
    @c_middle char(2),
```

```
    @c_street_1 char(20),
```

```
    @c_street_2 char(20),
```

```
    @c_city char(20),
```

```
    @c_state char(2),
```

```
    @c_zip char(9),
```

```
    @c_phone char(16),
```

```
    @c_since datetime,
```

```
    @c_credit char(2),
```

```
    @c_credit_lim numeric(12,2),
```

```
    @c_balance numeric(12,2),
```

```
    @c_discount numeric(4,4),
```

```
    @data char(500),
```

```
    @c_data char(500),
```

```
    @datetime datetime,
```

```
    @w_ytd numeric(12,2),
```

```
@d_ytd      numeric(12,2),
```

```
@cnt        smallint,
```

```
@val        smallint,
```

```
@screen_data char(200),
```

```
    @d_id_local tinyint,
```

```
    @w_id_local smallint,
```

```
    @c_id_local int
```

```
declare @delaytime varchar(30)
```

```
-- uniform random delay of 0 - 0.3 second; avg = 0.15
```

```
select @delaytime = '00:00:0' + cast(cast((rand()*0.30) 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 proc tpcc_stocklevel      @w_id          smallint,
                                @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()*3.60) as decimal(4,3)) as
char(5))
waitfor delay @delaytime

select 49

GO

create proc 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] [smallint] 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

```

Appendix B: Database Design

Database Build

backup.sql

```
-- File:  BACKUP.SQL
--      Microsoft TPC-C Benchmark Kit Ver. 4.22
--      Copyright Microsoft, 2001
-- Purpose: Creates backup of tpcc database

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

dump database tpcc to tpccback1, tpccback2, tpccback3, tpccback4 with init,
stats = 1

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

backupdev.sql

```
-- File:  BACKUPDEV.BSQL
--      Microsoft TPC-C Benchmark Kit Ver. 4.22
--      Copyright Microsoft, 2001
-- Purpose: Creates tpcc database Backup Devices

use master
go

-- create backup devices

exec sp_addumpdevice 'disk','tpccback1','Y:\tpccback1.dmp'
go
exec sp_addumpdevice 'disk','tpccback2','Z:\tpccback2.dmp'
go
exec sp_addumpdevice 'disk','tpccback3','Y:\tpccback3.dmp'
go
exec sp_addumpdevice 'disk','tpccback4','Z:\tpccback4.dmp'
go
```

createdb.sql

```
-- File:  CREATEDB.SQL
--      Microsoft TPC-C Benchmark Kit Ver. 4.22
--      Copyright Microsoft, 2001
-- Purpose: Creates tpcc database and backup files

use master
go

--      Create temporary table for timing

if exists ( select name from sysobjects where name = 'tpcc_timer' )
drop table tpcc_timer
```

```
go

create table tpcc_timer
(
        start_date          char(30),
        end_date            char(30)
)

insert      into tpcc_timer values (0,0)
go

--      Store starting time

update      tpcc_timer
set         start_date = (select convert(char(30), getdate(),9))
go

-- create main database files

CREATE DATABASE tpcc
ON PRIMARY
(
        NAME              = MSSQL_tpcc_root,
        FILENAME           = "C:\MSSQL_tpcc_root.mdf",
        SIZE               = 8MB,
        FILEGROWTH          =0),
FILEGROUP    MSSQL_misc_fg
(
        NAME              = MSSQL_misc1,
        FILENAME           = "H:",
        SIZE               = 100000MB,
        FILEGROWTH          = 0),
FILEGROUP    MSSQL_cs_fg
(
        NAME              = MSSQL_cs1,
        FILENAME           = "E:",
        SIZE               = 80000MB,
        FILEGROWTH          = 0),
(
        NAME              = MSSQL_cs2,
        FILENAME           = "F:",
        SIZE               = 80000MB,
        FILEGROWTH          = 0),
(
        NAME              = MSSQL_cs3,
        FILENAME           = "G:",
        SIZE               = 80000MB,
        FILEGROWTH          = 0)
LOG ON
(
        NAME              =MSSQL_tpcc_log,
        FILENAME           ="I:",
        SIZE               =100000MB,
        FILEGROWTH          =0)
COLLATE Latin1_General_BIN
go

-- Store ending time
update      tpcc_timer
set         end_date  = (select convert(char(30), getdate(),9))
go

select "Elapsed time (in seconds): ", datediff(second,(select start_date from
tpcc_timer),(select end_date from tpcc_timer))

--      remove temporary table

if exists ( select name from sysobjects where name = 'tpcc_timer' )
drop table tpcc_timer

go
```

dbopt1.sql

```
-- File:  DBOPT1.SQL
--      Microsoft TPC-C Benchmark Kit Ver. 4.22
--      Copyright Microsoft, 2001
-- Purpose: Sets database options for data load
```

```
use master
go
```

```
exec sp_dboption tpcc,'select into/bulkcopy',true
exec sp_dboption tpcc,'trunc. log on chkpt.',true
exec sp_dboption tpcc,'torn page detection',false
go
```

```
use tpcc
go
```

```
checkpoint
go
```

dbopt2.sql

```
-- File:  DBOPT2.SQL
--      Microsoft TPC-C Benchmark Kit Ver. 4.22
--      Copyright Microsoft, 2001
-- Purpose: Resets database options after data load
```

```
sp_dboption tpcc,'select into/bulkcopy',FALSE
GO
```

```
sp_dboption tpcc,'trunc. log on chkpt.',FALSE
GO
```

```
USE tpcc
GO
```

```
CHECKPOINT
GO
```

```
sp_configure 'allow updates',1
GO
```

```
RECONFIGURE WITH OVERRIDE
GO
```

```
DECLARE          @msg          varchar(50)
```

```
--
--      OPTIONS FOR SQL SERVER 8.0      --
-- 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
TRUE
EXEC sp_indexoption      'stock',    'DisallowPageLocks', TRUE
```

```
EXEC sp_indexoption      'order_line',      'DisallowRowLocks',
TRUE
EXEC sp_indexoption      'orders',    'DisallowRowLocks', TRUE
EXEC sp_indexoption      'new_order',    'DisallowRowLocks',
TRUE
EXEC sp_indexoption      'item',        'DisallowRowLocks',
TRUE
EXEC sp_indexoption      'item',        'DisallowPageLocks',
TRUE
GO
```

```
Print ''
Print '*****'
Print 'Pre-specified Locking Hierarchy:'
Print '      Lockflag = 0 ==> No pre-specified hierarchy'
Print '      Lockflag = 1 ==> Lock at Page-level then Table-level'
Print '      Lockflag = 2 ==> Lock at Row-level then Table-level'
Print '      Lockflag = 3 ==> Lock at Table-level'
Print ''
```

```
SELECT  name,lockflags
FROM      sysindexes
WHERE     object_id('warehouse')      = id OR
          object_id('district')      = id OR
          object_id('customer')      = id OR
          object_id('stock')          = id OR
          object_id('orders')          = id OR
          object_id('order_line')      = id OR
          object_id('history')          = id OR
          object_id('new_order')      = id OR
          object_id('item')            = id
ORDER BY  lockflags asc
GO
```

```
sp_configure 'allow updates',0
GO
```

```
RECONFIGURE WITH OVERRIDE
GO
```

```
EXEC sp_dboption tpcc,      'auto update statistics',FALSE
EXEC sp_dboption tpcc,      'auto create statistics', FALSE
GO
```

```
EXEC sp_tableoption 'district', 'pintable',true
EXEC sp_tableoption 'warehouse', 'pintable',true
EXEC sp_tableoption 'new_order', 'pintable',true
EXEC sp_tableoption 'item',      'pintable',true
GO
```

idxcuscl.sql

```
-- File:  IDXCUSCL.SQL
--      Microsoft TPC-C Benchmark Kit Ver. 4.22
--      Copyright Microsoft, 2001
-- Purpose: Creates clustered index on customer table
```

```
use tpcc
go
```

```
declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)
```

```
if exists ( select name from sysindexes where name = 'customer_cl' )
```

```

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,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxcusnc.sql

```

-- File:  IDXCUSNC.SQL
--      Microsoft TPC-C Benchmark Kit Ver. 4.22
--      Copyright Microsoft, 2001
-- Purpose: Creates non-clustered index on customer table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'customer_nc1' )
drop index customer.customer_nc1

create unique nonclustered index customer_nc1 on customer(c_w_id, c_d_id,
c_last, c_first, c_id)
on MSSQL_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxdiscl.sql

```

-- File:  IDXDISCL.SQL
--      Microsoft TPC-C Benchmark Kit Ver. 4.22
--      Copyright Microsoft, 2001
-- Purpose: Creates clustered index on district table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'district_c1' )
drop index district.district_c1

create unique clustered index district_c1 on district(d_w_id, d_id)
with fillfactor=100 on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

```

go

idxitmcl.sql

```

-- File:  IDXITMCL.SQL
--      Microsoft TPC-C Benchmark Kit Ver. 4.22
--      Copyright Microsoft, 2001
-- Purpose: Creates clustered index on item table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'item_c1' )
drop index item.item_c1

create unique clustered index item_c1 on item(i_id)
on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxnodcl.sql

```

-- File:  IDXNODCL.SQL
--      Microsoft TPC-C Benchmark Kit Ver. 4.22
--      Copyright Microsoft, 2001
-- Purpose: Creates clustered index on new_order table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'new_order_c1' )
drop index new_order.new_order_c1

create unique clustered index new_order_c1 on new_order(no_w_id, no_d_id,
no_o_id)
on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxodlcl.sql

```

-- File:  IDXODLCL.SQL

```

```
-- Microsoft TPC-C Benchmark Kit Ver. 4.22
-- Copyright Microsoft, 2001
-- Purpose: Creates clustered index on order_line table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'order_line_c1' )
    drop index order_line.order_line_c1

create unique clustered index order_line_c1 on order_line(ol_w_id, ol_d_id,
ol_o_id, ol_number)
    on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxordcl.sql

```
-- File: IDXORDCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.22
-- Copyright Microsoft, 2001
-- Purpose: Creates clustered index on orders table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'orders_c1' )
    drop index orders.orders_c1

create unique clustered index orders_c1 on orders(o_w_id, o_d_id, o_id)
    on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxordnc.sql

```
-- File: IDXORDNC.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.22
-- Copyright Microsoft, 2001
-- Purpose: Creates non-clustered index on orders table

use tpcc
go
```

```
declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'orders_nc1' )
    drop index orders.orders_nc1

create index orders_nc1 on orders(o_w_id, o_d_id, o_c_id, o_id)
    on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxstkcl.sql

```
-- File: IDXSTKCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.22
-- Copyright Microsoft, 2001
-- Purpose: Creates clustered index on stock table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'stock_c1' )
    drop index stock.stock_c1

create unique clustered index stock_c1 on stock(s_i_id, s_w_id)
    on MSSQL_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxwarcl.sql

```
-- File: IDXWARCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.22
-- Copyright Microsoft, 2001
-- Purpose: Creates clustered index on warehouse table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'warehouse_c1' )
    drop index warehouse.warehouse_c1

create unique clustered index warehouse_c1 on warehouse(w_id)
```

```
with fillfactor=100 on MSSQL_misc_fg
```

```
select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)
```

```
go
```

removedb.sql

```
-- File:   REMOVEDB.SQL
--        Microsoft TPC-C Benchmark Kit Ver. 4.22
--        Copyright Microsoft, 2001
-- Purpose: Removes tpcc database and backup files
```

```
use master
go
```

```
-- remove any existing database and backup files
```

```
exec sp_dbremove tpcc, dropdev
go
```

```
exec sp_dropdevice 'tpccback1'
exec sp_dropdevice 'tpccback2'
exec sp_dropdevice 'tpccback3'
exec sp_dropdevice 'tpccback4'
go
```

restore.sql

```
-- File:   RESTORE.SQL
--        Microsoft TPC-C Benchmark Kit Ver. 4.22
--        Copyright Microsoft, 2001
-- Purpose: Loads database backup from backup files
```

```
declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
--select "Start date:", convert(varchar(30),@startdate,9)
```

```
load database tpcc from tpccback1, tpccback2, tpccback3, tpccback4 with stats
= 1
```

```
select @enddate = getdate()
--select "End date: ", convert(varchar(30),@enddate,9)
--select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)
```

```
go
```

RunSQLCfg.sql

```
/* TPC-C Benchmark Kit */
/* */
/* RUNSQLCFG.SQL */
/* */
/* This script file is used to set runtime server configuration parameters */
/* */
```

```
exec sp_configure "show advanced option", 1
```

```
go
```

```
reconfigure with override
```

```
go
```

```
/* change this value to approximately the number of connected users */
exec sp_configure "max worker threads",240
```

```
/* increase priority of user threads */
exec sp_configure "priority boost",1
```

```
/* disable automatic checkpointing */
exec sp_configure "recovery interval",56
```

```
/* change to a mask appropriate for the number of processors on the server */
exec sp_configure "affinity mask",0x7
```

```
/* enable fibers */
exec sp_configure "lightweight pooling",1
```

```
/* enable update */
exec sp_configure "allow updates",1
```

```
/* set max degree of parallelism */
exec sp_configure "max degree of parallelism",1
```

```
go
```

```
reconfigure with override
```

```
go
```

sqlshutdown.sql

```
use tpcc
go
checkpoint
go
shutdown
go
```

tables.sql

```
-- File:   TABLES.SQL
--        Microsoft TPC-C Benchmark Kit Ver. 4.22
--        Copyright Microsoft, 2001
-- Purpose: Creates TPC-C tables
```

```
use tpcc
go
```

```
--
-- Remove all existing TPC-C tables
--
```

```
if exists ( select name from sysobjects where name = 'warehouse' )
drop table warehouse
```

```
go
if exists ( select name from sysobjects where name = 'district' )
drop table district
```

```
go
if exists ( select name from sysobjects where name = 'customer' )
drop table customer
```

```
go
if exists ( select name from sysobjects where name = 'history' )
```

```

drop table history
go
if exists ( select name from sysobjects where name = 'new_order' )
drop table new_order
go
if exists ( select name from sysobjects where name = 'orders' )
drop table orders
go
if exists ( select name from sysobjects where name = 'order_line' )
drop table order_line
go
if exists ( select name from sysobjects where name = 'item' )
drop table item
go
if exists ( select name from sysobjects where name = 'stock' )
drop table stock
go
--
-- Create new tables
--

create table warehouse
(
    w_id                smallint,
    w_name              char(10),
    w_street_1          char(20),
    w_street_2          char(20),
    w_city              char(20),
    w_state             char(2),
    w_zip              char(9),
    w_tax              numeric(4,4),
    w_ytd              numeric(12,2)
) on MSSQL_misc_fg
go

create table district
(
    d_id                tinyint,
    d_w_id              smallint,
    d_name              char(10),
    d_street_1          char(20),
    d_street_2          char(20),
    d_city              char(20),
    d_state             char(2),
    d_zip              char(9),
    d_tax              numeric(4,4),
    d_ytd              numeric(12,2),
    d_next_o_id         int
) on MSSQL_misc_fg
go

create table customer
(
    c_id                int,
    c_d_id              tinyint,
    c_w_id              smallint,
    c_first             char(16),
    c_middle            char(2),
    c_last             char(16),
    c_street_1          char(20),
    c_street_2          char(20),
    c_city              char(20),
    c_state             char(2),
    c_zip              char(9),
    c_phone            char(16),
    c_since            datetime,
    c_credit            char(2),

```

```

    c_credit_lim        numeric(12,2),
    c_discount          numeric(4,4),
    c_balance          numeric(12,2),
    c_ytd_payment      numeric(12,2),
    c_payment_cnt      smallint,
    c_delivery_cnt      smallint,
    c_data             char(500)
) on MSSQL_cs_fg
go

create table history
(
    h_c_id              int,
    h_c_d_id            tinyint,
    h_c_w_id            smallint,
    h_d_id              tinyint,
    h_w_id              smallint,
    h_date              datetime,
    h_amount            numeric(6,2),
    h_data             char(24)
) on MSSQL_misc_fg
go

create table new_order
(
    no_o_id             int,
    no_d_id             tinyint,
    no_w_id             smallint
) on MSSQL_misc_fg
go

create table orders
(
    o_id                int,
    o_d_id              tinyint,
    o_w_id              smallint,
    o_c_id              int,
    o_entry_d           datetime,
    o_carrier_id        tinyint,
    o_ol_cnt            tinyint,
    o_all_local         tinyint
) on MSSQL_misc_fg
go

create table order_line
(
    ol_o_id             int,
    ol_d_id             tinyint,
    ol_w_id             smallint,
    ol_number           tinyint,
    ol_i_id             int,
    ol_supply_w_id      smallint,
    ol_delivery_d        datetime,
    ol_quantity         smallint,
    ol_amount           numeric(6,2),
    ol_dist_info        char(24)
) on MSSQL_misc_fg
go

create table item
(
    i_id                int,
    i_im_id             int,
    i_name              char(24),
    i_price             numeric(5,2),
    i_data             char(50)
) on MSSQL_misc_fg
go

```

```

create table stock
(
    s_i_id                int,
    s_w_id                smallint,
    s_quantity            smallint,
    s_dist_01             char(24),
    s_dist_02             char(24),
    s_dist_03             char(24),
    s_dist_04             char(24),
    s_dist_05             char(24),
    s_dist_06             char(24),
    s_dist_07             char(24),
    s_dist_08             char(24),
    s_dist_09             char(24),
    s_dist_10             char(24),
    s_ytd                 int,
    s_order_cnt            smallint,
    s_remote_cnt          smallint,
    s_data                char(50)
) on MSSQL_cs_fg
go

```

Verify_TpccLoad.sql

```

-- File:  VERIFYTPCCLOAD.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.22
-- Copyright Microsoft, 2001
-- Purpose: Performs series of TPCC database checks to verify
--          that database load completed correctly

print  " "
select  convert(char(30), getdate(),9)
print  " "

use tpcc
go

-- *****
--
-- Check rows per table from SYSINDEXES
--
-- *****

print  'WAREHOUSE TABLE'

select  rows
from    sysindexes
where   id      = object_id("warehouse")
go

print  'DISTRICT TABLE = (10 * No of warehouses)'

select  rows
from    sysindexes
where   id      =object_id("district")
go

print  'ITEM TABLE = 100,000'

select  rows
from    sysindexes
where   id      =object_id("item")
go

print  'CUSTOMER TABLE = (30,000 * No of warehouses)'

```

```

select  rows
from    sysindexes
where   id      =object_id("customer")
go

print 'ORDERS TABLE = (30,000 * No of warehouses)'

select  rows
from    sysindexes
where   id      =object_id("orders")
go

print  'HISTORY TABLE = (30,000 * No of warehouses)'

select  rows
from    sysindexes
where   id      =object_id("history")
go

print  'STOCK TABLE = (100,000 * No of warehouses)'

select  rows
from    sysindexes
where   id      =object_id("stock")
go

print  'ORDER_LINE TABLE = (300,000 * No of warehouses + some
change)'

select  rows
from    sysindexes
where   id      =object_id("order_line")
go

print  'NEW_ORDER TABLE = (9000 * No of warehouses)'

select  rows
from    sysindexes
where   id      =object_id("new_order")
go

-- *****
--
-- Check indices
--
-- *****

print '*****Index Check*****'

use tpcc
go

sp_helpindex      customer
go

sp_helpindex      stock
go

sp_helpindex      district
go

sp_helpindex      item
go

sp_helpindex      new_order
go

sp_helpindex      orders

```

```

go

sp_helpindex      order_line
go

sp_helpindex      warehouse
go

```

version.sql

```

-- File:  VERSION.SQL
--      Microsoft TPC-C Benchmark Kit Ver. 4.22
--      Copyright Microsoft, 2001
-- Purpose: Returns version level of TPC-C stored procs
-- Note:   Always update the return value of this proc for
--         any interface changes or "must have" bug fixes.
--
-- The value returned by this SP defines the "interface level",
-- which must match between the stored procs and the client code.
-- The interface level may be down rev from the current kit. This
-- indicates that the interface hasn't changed since that version.

use tpcc
go

if exists ( select name from sysobjects where name = "tpcc_version" )
    drop procedure tpcc_version
go

create proc tpcc_version
as
declare    @version char(8)

begin
    select @version = "4.10.000"
    select @version as "Version"
end

go

```

Load Source Code

getargs.c

```

//      File:          GETARGS.C
//      Microsoft TPC-C Kit Ver. 4.22
//      Copyright Microsoft, 1996, 1997,
//      1998, 1999, 2000, 2001
//      Purpose:   Source file for command line processing

// Includes
#include "tpcc.h"

//=====
//
// Function name: GetArgsLoader
//
//=====

void GetArgsLoader(int argc, char **argv, TPCCLDR_ARGS *pargs)
{

```

```

    int          i;
    char        *ptr;

#ifdef DEBUG
    printf("[%ld]DBG: Entering GetArgsLoader()\n", (int)
GetCurrentThreadId());
#endif

    /* init args struct with some useful values */
    pargs->server      = SERVER;
    pargs->user        = USER;
    pargs->password     = PASSWORD;
    pargs->database     = DATABASE;
    pargs->batch        = BATCH;
    pargs->num_warehouses = UNDEF;
    pargs->tables_all   = TRUE;
    pargs->table_item   = FALSE;
    pargs->table_warehouse = FALSE;
    pargs->table_customer = FALSE;
    pargs->table_orders  = FALSE;
    pargs->loader_res_file =
    LOADER_RES_FILE;
    pargs->pack_size    = DEFLDPACKSIZE;
    pargs->starting_warehouse =
    DEF_STARTING_WAREHOUSE;
    pargs->build_index  = BUILD_INDEX;
    pargs->index_order  = INDEX_ORDER;
    pargs->index_script_path = INDEX_SCRIPT_PATH;
    pargs->scale_down   = SCALE_DOWN;

    /* check for zero command line args */
    if ( argc == 1 )
        GetArgsLoaderUsage();

    for (i = 1; i < argc; ++i)
    {
        if (argv[i][0] != '-' && argv[i][0] != '/')
        {
            printf("\nUnrecognized command");
            GetArgsLoaderUsage();
            exit(1);
        }

        ptr = argv[i];

        switch (ptr[1])
        {
            case 'h':    /* Fall through */
            case 'H':
                GetArgsLoaderUsage();
                break;

            case 'D':
                pargs->database = ptr+2;
                break;

            case 'P':
                pargs->password = ptr+2;
                break;

            case 'S':
                pargs->server = ptr+2;
                break;

            case 'U':
                pargs->user = ptr+2;
                break;

```

	case 'b':	pargs->batch = atol(ptr+2); break;	default:	GetArgsLoaderUsage(); exit(-1); break;
atol(ptr+2);	case 'W':	pargs->num_warehouses = break;	}	
atol(ptr+2);	case 's':	pargs->starting_warehouse = break;	/* check for required args */ if (pargs->num_warehouses == UNDEF) { printf("Number of Warehouses is required\n"); exit(-2); }	
FALSE; == 0)	case 't':	{ pargs->tables_all = if (strcmp(ptr+2,"item") else if else if else if else { printf("\nUnrecognized GetArgsLoaderUsage(); exit(1); } break; }	return; }	//=====
pargs->table_item = TRUE; (strcmp(ptr+2,"warehouse") == 0) pargs->table_warehouse = TRUE; (strcmp(ptr+2,"customer") == 0) pargs->table_customer = TRUE; (strcmp(ptr+2,"orders") == 0) pargs->table_orders = TRUE;			// // Function name: GetArgsLoaderUsage // //=====	
command");			void GetArgsLoaderUsage() { #ifdef DEBUG printf("[%ld]DBG: Entering GetArgsLoaderUsage()\n", (int) GetCurrentThreadId()); #endif	
			printf("TPCCLDLDR:\n\n"); printf("Parameter Default\n"); printf("-----\n"); printf("-W Number of Warehouses to Load Required \n"); printf("-S Server %s\n", SERVER); printf("-U Username %s\n", USER); printf("-P Password %s\n", PASSWORD); printf("-D Database %s\n", DATABASE); printf("-b Batch Size %ld\n", (long)	
	case 'f':	pargs->loader_res_file = ptr+2; break;	BATCH); printf("-p TDS packet size %ld\n", (long)	
	case 'p':	pargs->pack_size = atol(ptr+2); break;	DEFLDPACKSIZE); printf("-f Loader Results Output Filename %s\n", LOADER_RES_FILE); printf("-s Starting Warehouse %ld\n", (long)	
	case 'i':	pargs->build_index = atol(ptr+2); break;	DEF_STARTING_WAREHOUSE); printf("-i Build Option (data = 0, data and index = 1) %ld\n", (long) BUILD_INDEX); printf("-o Cluster Index Build Order (before = 1, after = 0) %ld\n", (long) INDEX_ORDER); printf("-c Build Scaled Database (normal = 0, tiny = 1) %ld\n", (long) SCALE_DOWN);	
	case 'o':	pargs->index_order = atol(ptr+2); break;	printf("-d Index Script Path %s\n", INDEX_SCRIPT_PATH); printf("-t Table to Load all tables \n");	
	case 'c':	pargs->scale_down = atol(ptr+2); break;	printf(" [item warehouse customer orders]\n"); printf(" Notes: \n"); printf(" - the '-t' parameter may be included multiple times to \n"); printf(" specify multiple tables to be loaded \n"); printf(" - 'item' loads ITEM table \n"); printf(" - 'warehouse' loads WAREHOUSE, DISTRICT, and STOCK tables \n");	
	case 'd':	pargs->index_script_path = ptr+2; break;		

```

printf(" - 'customer' loads CUSTOMER and HISTORY tables\n");
printf(" - 'orders' load NEW-ORDER, ORDERS, ORDER-LINE tables\n");

printf("\nNote: Command line switches are case sensitive.\n");

exit(0);
}

```

random.c

```

//      File:          RANDOM.C
//      Microsoft TPC-C Kit Ver. 4.22
//      Copyright Microsoft, 1996, 1997,
//      1998, 1999, 2000, 2001
//      Purpose:   Random number generation routines for database loader

// Includes
#include "tpcc.h"
#include "math.h"

// Defines
#define A      16807
#define M      2147483647
#define Q      127773 /* M div A */
#define R      2836 /* M mod A */
#define Thread __declspec(thread)

// Globals
long      Thread Seed = 0; /* thread local seed */

/*****
 *
 * random -
 *   Implements a GOOD pseudo random number generator. This generator
 *   will/should? run the complete period before repeating.
 *
 * Copied from:
 *   Random Numbers Generators: Good Ones Are Hard to Find.
 *   Communications of the ACM - October 1988 Volume 31 Number 10
 *
 * Machine Dependencies:
 *   long must be 2 ^ 31 - 1 or greater.
 *
 *****/

/*****
 *
 * seed - load the Seed value used in irand and drand. Should be used before *
 *   first call to irand or drand.
 *
 *****/

void seed(long val)
{
#ifdef DEBUG
printf("[%d]DBG: Entering seed()...\n", (int) GetCurrentThreadId());
printf("Old Seed %ld New Seed %ld\n", Seed, val);
#endif

if ( val < 0 )

```

```

    val = abs(val);

    Seed = val;
}

/*****
 *
 * irand - returns a 32 bit integer pseudo random number with a period of *
 *   1 to 2 ^ 32 - 1.
 *
 * parameters:
 *   none.
 *
 * returns:
 *   32 bit integer - defined as long ( see above ).
 *
 * side effects:
 *   seed get recomputed.
 *****/

long irand()
{
    register long  s; /* copy of seed */
    register long  test; /* test flag */
    register long  hi; /* tmp value for speed */
    register long  lo; /* tmp value for speed */

#ifdef DEBUG
printf("[%d]DBG: Entering irand()...\n", (int) GetCurrentThreadId());
#endif

    s = Seed;
    hi = s / Q;
    lo = s % Q;

    test = A * lo - R * hi;
    if ( test > 0 )
        Seed = test;
    else
        Seed = test + M;

    return( Seed );
}

/*****
 *
 * drand - returns a double pseudo random number between 0.0 and 1.0. *
 *   See irand.
 *****/

double drand()
{
#ifdef DEBUG
printf("[%d]DBG: Entering drand()...\n", (int) GetCurrentThreadId());
#endif

    return( (double)irand() / 2147483647.0 );
}

```

```
//=====
// Function : RandomNumber
//
// Description:
//=====

long RandomNumber(long lower, long upper)
{
    long rand_num;

#ifdef DEBUG
    printf("[%d]DBG: Entering RandomNumber()...\n", (int)
GetCurrentThreadId());
#endif

    if ( upper == lower ) /* pgd 08-13-96 perf enhancement */
        return lower;

    upper++;

    if ( upper <= lower )
        rand_num = upper;
    else
        rand_num = lower + irand() % (upper - lower); /* pgd
08-13-96 perf enhancement */

#ifdef DEBUG
    printf("[%d]DBG: RandomNumber between %ld & %ld ==> %ld\n",
(int) GetCurrentThreadId(), lower,
upper, rand_num);
#endif

    return rand_num;
}

#if 0

//Original code pgd 08/13/96

long RandomNumber(long lower,
long upper)
{
    long rand_num;

#ifdef DEBUG
    printf("[%d]DBG: Entering RandomNumber()...\n", (int)
GetCurrentThreadId());
#endif

    upper++;

    if ((upper <= lower))
        rand_num = upper;
    else
        rand_num = lower + irand() % ((upper > lower) ? upper -
lower : upper);

#ifdef DEBUG
    printf("[%d]DBG: RandomNumber between %ld & %ld ==> %ld\n",
(int) GetCurrentThreadId(), lower,
upper, rand_num);
#endif
#endif

```

```
        return rand_num;
    }
#endif

//=====
// Function : NURand
//
// Description:
//=====

long NURand(int iConst,
long x,
long y,
long C)
{
    long rand_num;

#ifdef DEBUG
    printf("[%d]DBG: Entering NURand()...\n", (int) GetCurrentThreadId());
#endif

    rand_num = (((RandomNumber(0,iConst) | RandomNumber(x,y)) + C) %
(y-x+1))+x;

#ifdef DEBUG
    printf("[%d]DBG: NURand: num = %d\n", (int) GetCurrentThreadId(),
rand_num);
#endif

    return rand_num;
}

```

strings.c

```
// File: STRINGS.C
// Microsoft TPC-C Kit Ver. 4.22
// Copyright Microsoft, 1996, 1997,
1998, 1999, 2000, 2001
// Purpose: Source file for database loader string functions

// Includes
#include "tpcc.h"
#include <string.h>
#include <ctype.h>

//=====
//
// Function name: MakeAddress
//
//=====

void MakeAddress(char *street_1,
char *street_2,
char *city,
char *state,
char *zip)
{
#ifdef DEBUG
    printf("[%d]DBG: Entering MakeAddress()\n", (int) GetCurrentThreadId());
#endif

```

```

    MakeAlphaString (10, 20, ADDRESS_LEN, street_1);
    MakeAlphaString (10, 20, ADDRESS_LEN, street_2);
    MakeAlphaString (10, 20, ADDRESS_LEN, city);
    MakeAlphaString ( 2, 2, STATE_LEN, state);
    MakeZipNumberString( 9, 9, ZIP_LEN, zip);

#ifdef DEBUG
    printf("[%ld]DBG: MakeAddress: street_1: %s, street_2: %s, city: %s, state:
%s, zip: %s\n",
                (int) GetCurrentThreadId(), street_1, street_2,
city, state, zip);
#endif

    return;
}

//=====
//
// Function name: LastName
//
//=====

void LastName(int num,
              char *name)
{
    static char *n[] =
    {
        "BAR", "OUGHT", "ABLE", "PRI", "PRES",
        "ESE", "ANTI", "CALLY", "ATION", "EING"
    };

#ifdef DEBUG
    printf("[%ld]DBG: Entering LastName()\n", (int) GetCurrentThreadId());
#endif

    if ((num >= 0) && (num < 1000))
    {
        strcpy(name, n[(num/100)%10]);
        strcat(name, n[(num/10)%10]);
        strcat(name, n[(num/1)%10]);

        if (strlen(name) < LAST_NAME_LEN)
        {
            PaddString(LAST_NAME_LEN, name);
        }
    }
    else
    {
        printf("\nError in LastName()... num <=%ld> out of range
(0,999)\n", num);
        exit(-1);
    }

#ifdef DEBUG
    printf("[%ld]DBG: LastName: num = [%d] ==> [%d][%d][%d]\n",
                (int) GetCurrentThreadId(), num, num/100,
(num/10)%10, num%10);
    printf("[%ld]DBG: LastName: String = %s\n", (int)
GetCurrentThreadId(), name);
#endif

    return;
}

```

```

}

//=====
//
// Function name: MakeAlphaString
//
//=====

//philipdu 08/13/96 Changed MakeAlphaString to use A-Z, a-z, and 0-9 in
//accordance with spec see below:
//The spec says:
//4.3.2.2 The notation random a-string [x .. y]
//(respectively, n-string [x .. y]) represents a string of random alphanumeric
//(respectively, numeric) characters of a random length of minimum x,
maximum y,
//and mean (y+x)/2. Alphanumerics are A..Z, a..z, and 0..9. The only other
//requirement is that the character set used "must be able to represent a
minimum
//of 128 different characters". We are using 8-bit chars, so this is a non issue.
//It is completely unreasonable to stuff non-printing chars into the text fields.
//CLevine 08/13/96

int MakeAlphaString( int x, int y, int z, char *str)
{
    int len;
    int i;
    char cc = 'a';
    static char chArray[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
    static int chArrayMax = 61;

#ifdef DEBUG
    printf("[%ld]DBG: Entering MakeAlphaString()\n", (int)
GetCurrentThreadId());
#endif

    len= RandomNumber(x, y);

    for (i=0; i<len; i++)
    {
        cc = chArray[RandomNumber(0, chArrayMax)];
        str[i] = cc;
    }

    if ( len < z )
        memset(str+len, ' ', z - len);
    str[len] = 0;

    return len;
}

//=====
//
// Function name: MakeOriginalAlphaString
//
//=====

int MakeOriginalAlphaString(int x,
                            int y,
                            int z,

```

<pre> char *str, int percent) { int len; int val; int start; #ifdef DEBUG printf("[%ld]DBG: Entering MakeOriginalAlphaString()\n", (int) GetCurrentThreadId()); #endif // verify percentage is valid if ((percent < 0) (percent > 100)) { printf("MakeOriginalAlphaString: Invalid percentage: %d\n", percent); exit(-1); } // verify string is at least 8 chars in length if ((x + y) <= 8) { printf("MakeOriginalAlphaString: string length must be >= 8\n"); exit(-1); } // Make Alpha String len = MakeAlphaString(x,y, z, str); val = RandomNumber(1,100); if (val <= percent) { start = RandomNumber(0, len - 8); strncpy(str + start, "ORIGINAL", 8); } #ifdef DEBUG printf("[%ld]DBG: MakeOriginalAlphaString: : %s\n", (int) GetCurrentThreadId(), str); #endif return strlen(str); } ===== // // Function name: MakeNumberString // ===== int MakeNumberString(int x, int y, int z, char *str) { char tmp[16]; //MakeNumberString is always called MakeZipNumberString(16, 16, 16, string) memset(str, '0', 16); itoa(RandomNumber(0, 99999999), tmp, 10); memcpy(str, tmp, strlen(tmp)); itoa(RandomNumber(0, 99999999), tmp, 10); memcpy(str+8, tmp, strlen(tmp)); </pre>	<pre> str[16] = 0; return 16; } ===== // // Function name: MakeZipNumberString // ===== int MakeZipNumberString(int x, int y, int z, char *str) { char tmp[16]; //MakeZipNumberString is always called MakeZipNumberString(9, 9, 9, string) strcpy(str, "000011111"); itoa(RandomNumber(0, 9999), tmp, 10); memcpy(str, tmp, strlen(tmp)); return 9; } ===== // // Function name: InitString // ===== void InitString(char *str, int len) { #ifdef DEBUG printf("[%ld]DBG: Entering InitString()\n", (int) GetCurrentThreadId()); #endif memset(str, '', len); str[len] = 0; } ===== // Function name: InitAddress // // Description: // ===== void InitAddress(char *street_1, char *street_2, char *city, char *state, char *zip) { memset(street_1, '', ADDRESS_LEN+1); memset(street_2, '', ADDRESS_LEN+1); memset(city, '', ADDRESS_LEN+1); street_1[ADDRESS_LEN+1] = 0; street_2[ADDRESS_LEN+1] = 0; city[ADDRESS_LEN+1] = 0; </pre>
--	--

```

        memset(state, '', STATE_LEN+1);
state[STATE_LEN+1] = 0;

        memset(zip, '', ZIP_LEN+1);
zip[ZIP_LEN+1] = 0;
}

//=====
//
// Function name: PaddString
//
//=====

void PaddString(int max, char *name)
{
    int        len;

    len = strlen(name);
    if ( len < max )
        memset(name+len, '', max - len);
    name[max] = 0;

    return;
}

```

time.c

```

//      File:                TIME.C
//
//      Microsoft TPC-C Kit Ver. 4.22
//      Copyright Microsoft, 1996, 1997,
//      1998, 1999, 2000, 2001
//      Purpose:   Source file for time functions

// Includes
#include "tpcc.h"

// Globals
static long start_sec;

//=====
//
// Function name: TimeNow
//
//=====

long TimeNow()
{
    long        time_now;
    struct      _timeb el_time;

#ifdef DEBUG
    printf("[%ld]DBG: Entering TimeNow()\n", (int) GetCurrentThreadId());
#endif

    _ftime(&el_time);

    time_now = ((el_time.time - start_sec) * 1000) + el_time.millitm;

    return time_now;
}

```

tpcc.h

```

//      File:                TPCC.H
//
//      Microsoft TPC-C Kit Ver. 4.22
//      Copyright Microsoft, 1996, 1997,
//      1998, 1999, 2000, 2001
//      Purpose:   Header file for TPC-C database loader

// Build number of TPC Benchmark Kit
#define TPCKIT_VER        "4.22"

// General headers
#include <windows.h>
#include <winbase.h>
#include <stdlib.h>
#include <stdio.h>
#include <process.h>
#include <stddef.h>
#include <stdarg.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <sys\types.h>

// ODBC headers
#include <sql.h>
#include <sqlext.h>
#include <odbc.h>

// General constants
#define MILLI            1000
#define FALSE            0
#define TRUE            1
#define UNDEF            -1
#define MINPRINTASCII    32
#define MAXPRINTASCII    126

// Default environment constants
#define SERVER            ""
#define DATABASE          "tpcc"
#define USER              "sa"
#define PASSWORD          ""

// Default loader arguments
#define BATCH            10000
#define DEFLDPACKSIZE    32768
#define LOADER_RES_FILE  "logs\load.out"
#define LOADER_NURAND_C  123
#define DEF_STARTING_WAREHOUSE 1
#define BUILD_INDEX      1
// build both data and indexes
#define INDEX_ORDER      1
// build indexes before load
#define SCALE_DOWN        0
// build a normal scale database
#define INDEX_SCRIPT_PATH "scripts"

typedef struct
{
    char        *server;
    char        *database;
    char        *user;
    char        *password;
    BOOL        tables_all;

// set if loading all tables

```

```

        BOOL                                table_item;
// set if loading ITEM table specifically
        BOOL                                table_warehouse;
set if loading WAREHOUSE, DISTRICT, and STOCK
        BOOL                                table_customer;
// set if loading CUSTOMER and HISTORY
        BOOL                                table_orders;
// set if loading NEW-ORDER, ORDERS, ORDER-LINE
long      num_warehouses;
long      batch;
long      verbose;

        long      pack_size;
        char      *loader_res_file;
        char      *synch_servername;
        long      case_sensitivity;
        long      starting_warehouse;
        long      build_index;
        long      index_order;
        long      scale_down;
        char      *index_script_path;
} TPCCLDR_ARGS;

// String length constants
#define SERVER_NAME_LEN      20
#define DATABASE_NAME_LEN    20
#define USER_NAME_LEN        20
#define PASSWORD_LEN         20
#define TABLE_NAME_LEN     20
#define I_DATA_LEN           50
#define I_NAME_LEN           24
#define BRAND_LEN            1
#define LAST_NAME_LEN        16
#define W_NAME_LEN           10
#define ADDRESS_LEN          20
#define STATE_LEN            2
#define ZIP_LEN              9
#define S_DIST_LEN           24
#define S_DATA_LEN           50
#define D_NAME_LEN           10
#define FIRST_NAME_LEN       16
#define MIDDLE_NAME_LEN      2
#define PHONE_LEN            16
#define CREDIT_LEN           2
#define C_DATA_LEN           500
#define H_DATA_LEN           24
#define DIST_INFO_LEN        24
#define MAX_OL_NEW_ORDER_ITEMS 15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN           25
#define OL_DIST_INFO_LEN     24
#define C_SINCE_LEN
23
#define H_DATE_LEN
23
#define OL_DELIVERY_D_LEN    23
#define O_ENTRY_D_LEN        23

// Functions in random.c
void      seed();
long      irand();
double    drand();
void      WUCreate();
short     WURand();
long      RandomNumber(long lower, long upper);

// Functions in getargs.c;
void      GetArgsLoader();

```

```
void      GetArgsLoaderUsage();
```

```
// Functions in time.c
```

```
long      TimeNow();
```

```
// Functions in strings.c
```

```
void      MakeAddress();
```

```
void      LastName();
```

```
int      MakeAlphaString();
```

```
int      MakeOriginalAlphaString();
```

```
int      MakeNumberString();
```

```
int      MakeZipNumberString();
```

```
void      InitString();
```

```
void      InitAddress();
```

```
void      PaddString();
```

tpccldr.c

```
// File: TPCCLDR.C
```

```
// Microsoft TPC-C Kit Ver. 4.22
```

```
// Copyright Microsoft, 2000, 2001
```

```
// Purpose: Source file for TPC-C database loader
```

```
// Includes
```

```
#include "tpcc.h"
```

```
#include "search.h"
```

```
// Defines
```

```
#define MAXITEMS      100000
```

```
#define MAXITEMS_SCALE_DOWN      100
```

```
#define CUSTOMERS_PER_DISTRICT 3000
```

```
#define CUSTOMERS_SCALE_DOWN 30
```

```
#define DISTRICT_PER_WAREHOUSE 10
```

```
#define ORDERS_PER_DISTRICT 3000
```

```
#define ORDERS_SCALE_DOWN 30
```

```
#define MAX_CUSTOMER_THREADS 2
```

```
#define MAX_ORDER_THREADS 3
```

```
#define MAX_MAIN_THREADS 4
```

```
// Functions declarations
```

```
void HandleErrorDBC(SQLHDBC hdbc1);
```

```
void CheckSQL();
```

```
void CheckDataBase();
```

```
long NURand();
```

```
void LoadItem();
```

```
void LoadWarehouse();
```

```
void Stock();
```

```
void District();
```

```
void LoadCustomer();
```

```
void CustomerBufInit();
```

```
void CustomerBufLoad();
```

```
void LoadCustomerTable();
```

```
void LoadHistoryTable();
```

```
void LoadOrders();
```

```
void OrdersBufInit();
```

```
void OrdersBufLoad();
```

```
void LoadOrdersTable();
```

```
void LoadNewOrderTable();
```

```
void LoadOrderLineTable();
```

```
void GetPermutation();
```

```

void CheckForCommit();
void OpenConnections();
void BuildIndex();
void FormatDate ();

// Shared memory structures

typedef struct
{
    long        ol;
    long        ol_i_id;
    short       ol_supply_w_id;
    short       ol_quantity;
    double      ol_amount;
    char        ol_dist_info[DIST_INFO_LEN+1];
    char
ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCT;

typedef struct
{
    long        o_id;
    short       o_d_id;
    short       o_w_id;
    long        o_c_id;
    short       o_carrier_id;
    short       o_ol_cnt;
    short       o_all_local;
    ORDER_LINE_STRUCT o_ol[15];
} ORDERS_STRUCT;

typedef struct
{
    long
short
short
char
c_first[FIRST_NAME_LEN+1];
char
c_middle[MIDDLE_NAME_LEN+1];
char
char
char
char
char
char
char
char
double
double
// fix to avoid ODBC float to numeric conversion problem.
// double
char
double
short
short
char
c_data[C_DATA_LEN+1];
double
char
} CUSTOMER_STRUCT;

typedef struct
{
    char
c_last[LAST_NAME_LEN+1];

```

```

//
//=====

int main(int argc, char **argv)
{
    DWORD      dwThreadID[MAX_MAIN_THREADS];
    HANDLE      hThread[MAX_MAIN_THREADS];
    FILE        *fLoader;
    char        buffer[255];
    int         i;

    for (i=0; i<MAX_MAIN_THREADS; i++)
        hThread[i] = NULL;

    printf("\n*****");
    printf("\n*                               *");
    printf("\n* Microsoft SQL Server           *");
    printf("\n*                               *");
    printf("\n* TPC-C BENCHMARK KIT: Database loader *");
    printf("\n* Version %s                    *", TPCKIT_VER);
    printf("\n*                               *");

    printf("\n*****\n");

    // process command line arguments

    aptr = &args;
    GetArgsLoader(argc, argv, aptr);

    // verify database and tables exist before attempting to load

    CheckSQL();
    CheckDataBase();

    printf("Build interface is ODBC.\n");

    if (aptr->build_index == 0)
        printf("Data load only - no index creation.\n");
    else
        printf("Data load and index creation.\n");

    if (aptr->index_order == 0)
        printf("Clustered indexes will be created after bulk
load.\n");
    else
        printf("Clustered indexes will be created before bulk
load.\n");

    // set database scale values
    if (aptr->scale_down == 1)
    {
        printf("*** Scaled Down Database ***\n");
        max_items = MAXITEMS_SCALE_DOWN;
        customers_per_district =
CUSTOMERS_SCALE_DOWN;
        orders_per_district = ORDERS_SCALE_DOWN;
        first_new_order = 0;
        last_new_order = 30;
    }
    else
    {
        max_items = MAXITEMS;
        customers_per_district =
CUSTOMERS_PER_DISTRICT;
        orders_per_district = ORDERS_PER_DISTRICT;
        first_new_order = 2100;

        last_new_order = 3000;
    }

    // open connections to SQL Server

    OpenConnections();

    // open file for loader results
    fLoader = fopen(aptr->loader_res_file, "w");

    if (fLoader == NULL)
    {
        printf("Error, loader result file open failed.");
        exit(-1);
    }

    // start loading data

    sprintf(buffer, "TPC-C load started for %ld
warehouses.\n", aptr->num_warehouses);

    printf("%s", buffer);
    fprintf(fLoader, "%s", buffer);

    main_time_start = (TimeNow() / MILLI);

    // start parallel load threads

    if (aptr->tables_all || aptr->table_item)
    {
        fprintf(fLoader, "\nStarting loader threads for: item\n");

        hThread[0] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadItem,
NULL,
0,
&dwThreadID[0]);

        if (hThread[0] == NULL)
        {
            printf("Error, failed in creating creating thread
= 0.\n");
            exit(-1);
        }
    }

    if (aptr->tables_all || aptr->table_warehouse)
    {
        fprintf(fLoader, "Starting loader threads for:
warehouse\n");

        hThread[1] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadWarehouse,
NULL,
0,

```

```

&dwThreadID[1]);

    if (hThread[1] == NULL)
    {
        printf("Error, failed in creating creating thread
= 1.\n");
        exit(-1);
    }

    if (aptr->tables_all || aptr->table_customer)
    {
        fprintf(fLoader, "Starting loader threads for:
customer\n");

        hThread[2] = CreateThread(NULL,

0,

(LPTHREAD_START_ROUTINE) LoadCustomer,

NULL,

0,

&dwThreadID[2]);

        if (hThread[2] == NULL)
        {
            printf("Error, failed in creating creating main
thread = 2.\n");
            exit(-1);
        }

        if (aptr->tables_all || aptr->table_orders)
        {
            fprintf(fLoader, "Starting loader threads for: orders\n");

            hThread[3] = CreateThread(NULL,

0,

(LPTHREAD_START_ROUTINE) LoadOrders,

NULL,

0,

&dwThreadID[3]);

            if (hThread[3] == NULL)
            {
                printf("Error, failed in creating creating main
thread = 3.\n");
                exit(-1);
            }

            // Wait for threads to finish...
            for (i=0; i<MAX_MAIN_THREADS; i++)
            {
                if (hThread[i] != NULL)
                {
                    WaitForSingleObject( hThread[i], INFINITE

);

                    CloseHandle(hThread[i]);

```

```

        hThread[i] = NULL;
    }

    main_time_end = (TimeNow() / MILLI);

    sprintf(buffer, "\nTPC-C load completed successfully in %ld minutes.\n",
        (main_time_end - main_time_start)/60);

    printf("%s", buffer);
    fprintf(fLoader, "%s", buffer);

    fclose(fLoader);

    SQLFreeEnv(henv);

    exit(0);

    return 0;
}

//=====
//
// Function name: LoadItem
//
//=====

void LoadItem()
{
    long        i_id;
    long        i_im_id;
    char        i_name[I_NAME_LEN+1];
    double      i_price;
    char        i_data[I_DATA_LEN+1];
    char        name[20];
    long        time_start;
    RETCODE     rc;
    DBINT       rcint;
    char        bcphint[128];

    // Seed with unique number
    seed(1);

    printf("Loading item table...\n");

    // if build index before load
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        BuildIndex("idxtmcl");

    InitString(i_name, I_NAME_LEN+1);
    InitString(i_data, I_DATA_LEN+1);

    sprintf(name, "%s..%s", aptr->database, "item");

    rc = bcp_init(i_hdbc1, name, NULL, "logs\\item.err", DB_IN);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcphint, "tablock, order (i_id),
ROWS_PER_BATCH = 100000");
        rc = bcp_control(i_hdbc1, BCPHINTS, (void*) bcphint);
        if (rc != SUCCEED)
            HandleErrorDBC(i_hdbc1);
    }
}

```

```

    }

    rc = bcp_bind(i_hdbc1, (BYTE *) &i_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 2);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    rc = bcp_bind(i_hdbc1, (BYTE *) i_name, 0, I_NAME_LEN,
NULL, 0, 0, 3);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    rc = bcp_bind(i_hdbc1, (BYTE *) &i_price, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 4);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0, I_DATA_LEN, NULL,
0, 0, 5);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    time_start = (TimeNow() / MILLI);

    item_rows_loaded = 0;

    for (i_id = 1; i_id <= max_items; i_id++)
    {
        i_im_id = RandomNumber(1L, 10000L);

        MakeAlphaString(14, 24, I_NAME_LEN, i_name);

        i_price = ((float) RandomNumber(100L, 10000L))/100.0;

        MakeOriginalAlphaString(26, 50, I_DATA_LEN, i_data,
10);

        rc = bcp_sendrow(i_hdbc1);
        if (rc != SUCCEED)
            HandleErrorDBC(i_hdbc1);

        item_rows_loaded++;
        CheckForCommit(i_hdbc1, i_hstmt1, item_rows_loaded,
"item", &time_start);
    }

    rcint = bcp_done(i_hdbc1);
    if (rcint < 0)
        HandleErrorDBC(i_hdbc1);

    printf("Finished loading item table.\n");

    SQLFreeStmt(i_hstmt1, SQL_DROP);
    SQLDisconnect(i_hdbc1);
    SQLFreeConnect(i_hdbc1);

    // if build index after load
    if ((aptr->build_index == 1) && (aptr->index_order == 0))
        BuildIndex("idxitmcl");
}

```

```

//=====
//
// Function : LoadWarehouse
//
// Loads WAREHOUSE table and loads Stock and District as Warehouses are
// created
//=====

void LoadWarehouse()
{
    short    w_id;
    char     w_name[W_NAME_LEN+1];
    char     w_street_1[ADDRESS_LEN+1];
    char     w_street_2[ADDRESS_LEN+1];
    char     w_city[ADDRESS_LEN+1];
    char     w_state[STATE_LEN+1];
    char     w_zip[ZIP_LEN+1];
    double   w_tax;
    double   w_ytd;
    char     name[20];
    long     time_start;
    RETCODE  rc;
    DBINT    rcint;
    char     bcphint[128];

    // Seed with unique number
    seed(2);

    printf("Loading warehouse table...\n");

    // if build index before load...
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        BuildIndex("idxwardcl");

    InitString(w_name, W_NAME_LEN+1);
    InitAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

    sprintf(name, "%s..%s", aptr->database, "warehouse");

    rc = bcp_init(w_hdbc1, name, NULL, "logs\\whouse.err", DB_IN);

    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcphint, "tablock, order (w_id,
ROWS_PER_BATCH = %d", aptr->num_warehouses);
        rc = bcp_control(w_hdbc1, BCPHINTS, (void*)
bcphint);

        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
    }

    rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0, W_NAME_LEN,
NULL, 0, 0, 2);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
}

```

<pre> rc = bcp_bind(w_hdbc1, (BYTE *) w_street_1, 0, ADDRESS_LEN, NULL, 0, 0, 3); if (rc != SUCCEED) HandleErrorDBC(w_hdbc1); rc = bcp_bind(w_hdbc1, (BYTE *) w_street_2, 0, ADDRESS_LEN, NULL, 0, 0, 4); if (rc != SUCCEED) HandleErrorDBC(w_hdbc1); rc = bcp_bind(w_hdbc1, (BYTE *) w_city, 0, ADDRESS_LEN, NULL, 0, 0, 5); if (rc != SUCCEED) HandleErrorDBC(w_hdbc1); rc = bcp_bind(w_hdbc1, (BYTE *) w_state, 0, STATE_LEN, NULL, 0, 0, 6); if (rc != SUCCEED) HandleErrorDBC(w_hdbc1); rc = bcp_bind(w_hdbc1, (BYTE *) w_zip, 0, ZIP_LEN, NULL, 0, 0, 7); if (rc != SUCCEED) HandleErrorDBC(w_hdbc1); rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 8); if (rc != SUCCEED) HandleErrorDBC(w_hdbc1); rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 9); if (rc != SUCCEED) HandleErrorDBC(w_hdbc1); time_start = (TimeNow() / MILLI); warehouse_rows_loaded = 0; for (w_id = (short)aptr->starting_warehouse; w_id <= aptr->num_warehouses; w_id++) { MakeAlphaString(6,10, W_NAME_LEN, w_name); MakeAddress(w_street_1, w_street_2, w_city, w_state, w_zip); w_tax = ((float) RandomNumber(0L,2000L))/10000.00; w_ytd = 300000.00; rc = bcp_sendrow(w_hdbc1); if (rc != SUCCEED) HandleErrorDBC(w_hdbc1); warehouse_rows_loaded++; CheckForCommit(w_hdbc1, i_hstmt1, warehouse_rows_loaded, "warehouse", &time_start); } rcint = bcp_done(w_hdbc1); if (rcint < 0) HandleErrorDBC(w_hdbc1); printf("Finished loading warehouse table.\n"); // if build index after load... if ((aptr->build_index == 1) && (aptr->index_order == 0)) </pre>	<pre> BuildIndex("idxwarc1"); stock_rows_loaded = 0; district_rows_loaded = 0; District(); Stock(); } //===== // // Function : District // //===== void District() { short d_id; short d_w_id; char d_name[D_NAME_LEN+1]; char d_street_1[ADDRESS_LEN+1]; char d_street_2[ADDRESS_LEN+1]; char d_city[ADDRESS_LEN+1]; char d_state[STATE_LEN+1]; char d_zip[ZIP_LEN+1]; double d_tax; double d_ytd; char name[20]; long d_next_o_id; long time_start; int w_id; RETCODE rc; DBINT rcint; char bcpint[128]; // Seed with unique number seed(4); printf("Loading district table...\n"); // build index before load if ((aptr->build_index == 1) && (aptr->index_order == 1)) BuildIndex("idxdiscl"); InitString(d_name, D_NAME_LEN+1); InitAddress(d_street_1, d_street_2, d_city, d_state, d_zip); sprintf(name, "%s..%s", aptr->database, "district"); rc = bcp_init(w_hdbc1, name, NULL, "logs\\district.err", DB_IN); if (rc != SUCCEED) HandleErrorDBC(w_hdbc1); if ((aptr->build_index == 1) && (aptr->index_order == 1)) { sprintf(bcpint, "tablock, order (d_w_id, d_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 10)); rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcpint); if (rc != SUCCEED) HandleErrorDBC(w_hdbc1); } } </pre>
---	---

```

        rc = bcp_bind(w_hdbc1, (BYTE *) &d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 1);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &d_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) d_name, 0, D_NAME_LEN,
NULL, 0, 0, 3);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) d_street_1, 0, ADDRESS_LEN,
NULL, 0, 0, 4);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) d_street_2, 0, ADDRESS_LEN,
NULL, 0, 0, 5);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) d_city, 0, ADDRESS_LEN,
NULL, 0, 0, 6);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) d_state, 0, STATE_LEN, NULL,
0, 0, 7);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) d_zip, 0, ZIP_LEN, NULL, 0, 0,
8);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &d_tax, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 9);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &d_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 10);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &d_next_o_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 11);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        d_ytd = 30000.0;

        d_next_o_id = orders_per_district+1;

        time_start = (TimeNow() / MILLI);

        for (w_id = aptr->starting_warehouse; w_id <=
aptr->num_warehouses; w_id++)
        {
            d_w_id = w_id;

            for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE;
d_id++)

```

```

        {
            MakeAlphaString(6,10,D_NAME_LEN,
d_name);

            MakeAddress(d_street_1, d_street_2, d_city,
d_state, d_zip);

            d_tax = ((float)
RandomNumber(0L,2000L))/10000.00;

            rc = bcp_sendrow(w_hdbc1);
            if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

            district_rows_loaded++;
            CheckForCommit(w_hdbc1, w_hstmt1,
district_rows_loaded, "district", &time_start);
        }
    }

    rcint = bcp_done(w_hdbc1);
    if (rcint < 0)
        HandleErrorDBC(w_hdbc1);

    printf("Finished loading district table.\n");

    // if build index after load...
    if ((aptr->build_index == 1) && (aptr->index_order == 0))
        BuildIndex("idxdiscl");

    return;
}

//=====
//
// Function : Stock
//
//=====

void Stock()
{
    long    s_i_id;
    short   s_w_id;
    short   s_quantity;
    char    s_dist_01[S_DIST_LEN+1];
    char    s_dist_02[S_DIST_LEN+1];
    char    s_dist_03[S_DIST_LEN+1];
    char    s_dist_04[S_DIST_LEN+1];
    char    s_dist_05[S_DIST_LEN+1];
    char    s_dist_06[S_DIST_LEN+1];
    char    s_dist_07[S_DIST_LEN+1];
    char    s_dist_08[S_DIST_LEN+1];
    char    s_dist_09[S_DIST_LEN+1];
    char    s_dist_10[S_DIST_LEN+1];
    long    s_ytd;
    short   s_order_cnt;
    short   s_remote_cnt;
    char    s_data[S_DATA_LEN+1];
    short   len;
    char    name[20];
    long    time_start;
    RETCODE rc;
    DBINT   rcint;
    char    bcphint[128];

```

```

// Seed with unique number
seed(3);

// if build index before load...
if ((aptr->build_index == 1) && (aptr->index_order == 1))
    BuildIndex("idxstkcl");

sprintf(name, "%s..%s", aptr->database, "stock");

rc = bcp_init(w_hdbc1, name, NULL, "logs\\stock.err", DB_IN);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    sprintf(bcphint, "tablock, order (s_i_id, s_w_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 100000));
    rc = bcp_control(w_hdbc1, BCPHINTS, (void*)
bcphint);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
}

rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 2);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_quantity, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01, 0, S_DIST_LEN,
NULL, 0, 0, 4);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02, 0, S_DIST_LEN,
NULL, 0, 0, 5);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03, 0, S_DIST_LEN,
NULL, 0, 0, 6);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04, 0, S_DIST_LEN,
NULL, 0, 0, 7);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05, 0, S_DIST_LEN,
NULL, 0, 0, 8);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06, 0, S_DIST_LEN,
NULL, 0, 0, 9);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

```

```

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07, 0, S_DIST_LEN,
NULL, 0, 0, 10);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0, S_DIST_LEN,
NULL, 0, 0, 11);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09, 0, S_DIST_LEN,
NULL, 0, 0, 12);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0, S_DIST_LEN,
NULL, 0, 0, 13);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 14);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_order_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 15);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_remote_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 16);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0, S_DATA_LEN,
NULL, 0, 0, 17);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

s_ytd = s_order_cnt = s_remote_cnt = 0;

time_start = (TimeNow()) / MILLI);

printf("...Loading stock table\n");

for (s_i_id=1; s_i_id <= max_items; s_i_id++)
{
    for (s_w_id = (short)aptr->starting_warehouse; s_w_id
<= aptr->num_warehouses; s_w_id++)
    {
        s_quantity =
(short)RandomNumber(10L,100L);
        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_01);
        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_02);
        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_03);
        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_04);
        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_05);
        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_06);

```

```

s_dist_07);
len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_08);
len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_09);
len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_10);
len = MakeAlphaString(24,24,S_DIST_LEN,

S_DATA_LEN, s_data,10);

rc = bcp_sendrow(w_hdbc1);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

stock_rows_loaded++;
CheckForCommit(w_hdbc1, w_hstmt1,
stock_rows_loaded, "stock", &time_start);

    }

rcint = bcp_done(w_hdbc1);
if (rcint < 0)
    HandleErrorDBC(w_hdbc1);

printf("Finished loading stock table.\n");

SQLFreeStmt(w_hstmt1, SQL_DROP);
SQLDisconnect(w_hdbc1);
SQLFreeConnect(w_hdbc1);

// if build index after load...
if ((aptr->build_index == 1) && (aptr->index_order == 0))
    BuildIndex("idxstkcl");

return;
}

//=====
//
// Function : LoadCustomer
//
//=====

void LoadCustomer()
{
    LOADER_TIME_STRUCT customer_time_start;
    LOADER_TIME_STRUCT history_time_start;
    short w_id;
    short d_id;
    DWORD
dwThreadId[MAX_CUSTOMER_THREADS];
HANDLE
hThread[MAX_CUSTOMER_THREADS];
char name[20];
RETCODE

rc;
DBINT rcint;
char
bcphint[128];
char cmd[256];
// SQLRETURN rc_1;

// SQLSMALLINT recnum,
MsgLen;
// SQLCHAR
SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
// SQLINTEGER NativeError;

// Seed with unique number
seed(5);

printf("Loading customer and history tables...\n");

// if build index before load...
if ((aptr->build_index == 1) && (aptr->index_order == 1))
    BuildIndex("idxcuscl");

// Initialize bulk copy
sprintf(name, "%s.%s", aptr->database, "customer");

rc = bcp_init(c_hdbc1, name, NULL, "logs\\customer.err", DB_IN);

if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    sprintf(bcpint, "tablock, order (c_w_id, c_d_id, c_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
rc = bcp_control(c_hdbc1, BCPHINTS, (void*) bcpint);

    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc1);
}

sprintf(name, "%s.%s", aptr->database, "history");

rc = bcp_init(c_hdbc2, name, NULL, "logs\\history.err", DB_IN);

if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

sprintf(bcpint, "tablock");
rc = bcp_control(c_hdbc2, BCPHINTS, (void*) bcpint);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

customer_rows_loaded = 0;
history_rows_loaded = 0;

CustomerBufInit();

customer_time_start.time_start = (TimeNow() / MILLI);
history_time_start.time_start = (TimeNow() / MILLI);

for (w_id = (short)aptr->starting_warehouse; w_id <=
aptr->num_warehouses; w_id++)
{
    for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE;
d_id++)
    {
        CustomerBufLoad(d_id, w_id);

        // Start parallel loading threads here...

        // Start customer table thread

        printf("...Loading customer table for: d_id =
%d, w_id = %d\n", d_id, w_id);
    }
}

```

```

        hThread[0] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadCustomerTable,
&customer_time_start,
0,
&dwThreadID[0]);

if (hThread[0] == NULL)
{
    printf("Error, failed in creating
creating thread = 0.\n");
    exit(-1);
}

// Start History table thread
printf("...Loading history table for: d_id = %d,
w_id = %d\n", d_id, w_id);

hThread[1] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadHistoryTable,
&history_time_start,
0,
&dwThreadID[1]);

if (hThread[1] == NULL)
{
    printf("Error, failed in creating
creating thread = 1.\n");
    exit(-1);
}

WaitForSingleObject( hThread[0], INFINITE
);
WaitForSingleObject( hThread[1], INFINITE
);

if (CloseHandle(hThread[0]) == FALSE)
{
    printf("Error, failed in closing
customer thread handle with errno: %d\n", GetLastError());
}

if (CloseHandle(hThread[1]) == FALSE)
{
    printf("Error, failed in closing
history thread handle with errno: %d\n", GetLastError());
}

}

// flush the bulk connection
rcint = bcp_done(c_hdbc1);

```

```

if (rcint < 0)
    HandleErrorDBC(c_hdbc1);

rcint = bcp_done(c_hdbc2);
if (rcint < 0)
    HandleErrorDBC(c_hdbc2);

printf("Finished loading customer table.\n");

// if build index after load...
if ((aptr->build_index == 1) && (aptr->index_order == 0))
    BuildIndex("idxcuscl");

// build non-clustered index
if (aptr->build_index == 1)
    BuildIndex("idxcusnc");

// Output the NURAND used for the loader into C_FIRST for C_ID
= 1,
// C_W_ID = 1, and C_D_ID = 1
sprintf(cmd, "isql -S%s -U%s -P%s -d%s -e -Q\"update customer set
c_first = 'C_LOAD = %d' where c_id = 1 and c_w_id = 1 and c_d_id = 1\" >
logs\\nurand_load.log",
aptr->server,
aptr->user,
aptr->password,
aptr->database,
LOADER_NURAND_C);

system(cmd);

SQLFreeStmt(c_hstmt1, SQL_DROP);
SQLDisconnect(c_hdbc1);
SQLFreeConnect(c_hdbc1);

SQLFreeStmt(c_hstmt2, SQL_DROP);
SQLDisconnect(c_hdbc2);
SQLFreeConnect(c_hdbc2);

return;
}

//=====
//
// Function : CustomerBufInit
//
//=====

void CustomerBufInit()
{
    int i;

    for (i=0;i<customers_per_district;i++)
    {
        customer_buf[i].c_id = 0;
        customer_buf[i].c_d_id = 0;
        customer_buf[i].c_w_id = 0;

        strcpy(customer_buf[i].c_first,"");
        strcpy(customer_buf[i].c_middle,"");
        strcpy(customer_buf[i].c_last,"");
        strcpy(customer_buf[i].c_street_1,"");
        strcpy(customer_buf[i].c_street_2,"");
    }
}

```

```

strcpy(customer_buf[i].c_city,"");
strcpy(customer_buf[i].c_state,"");
strcpy(customer_buf[i].c_zip,"");
strcpy(customer_buf[i].c_phone,"");
strcpy(customer_buf[i].c_credit,"");

customer_buf[i].c_credit_lim = 0;
customer_buf[i].c_discount = (float) 0;

// fix to avoid ODBC float to numeric conversion
problem.

//      customer_buf[i].c_balance = 0;
strcpy(customer_buf[i].c_balance,"");

customer_buf[i].c_ytd_payment = 0;
customer_buf[i].c_payment_cnt = 0;
customer_buf[i].c_delivery_cnt = 0;

strcpy(customer_buf[i].c_data,"");

customer_buf[i].h_amount = 0;

strcpy(customer_buf[i].h_data,"");

}

}

//=====
//
// Function : CustomerBufLoad
//
// Fills shared buffer for HISTORY and CUSTOMER
//=====

void CustomerBufLoad(int d_id, int w_id)
{
    long i;
    CUSTOMER_SORT_STRUCT c[CUSTOMERS_PER_DISTRICT];

    for (i=0;i<customers_per_district;i++)
    {
        if (i < 1000)
            LastName(i, c[i].c_last);
        else
            LastName(NURand(255,0,999,LOADER_NURAND_C), c[i].c_last);

        MakeAlphaString(8,16,FIRST_NAME_LEN,
c[i].c_first);

        c[i].c_id = i+1;
    }

    printf("...Loading customer buffer for: d_id = %d, w_id = %d\n",
        d_id, w_id);

    for (i=0;i<customers_per_district;i++)
    {
        customer_buf[i].c_d_id = d_id;
        customer_buf[i].c_w_id = w_id;

```

```

customer_buf[i].h_amount = 10.0;

customer_buf[i].c_ytd_payment = 10.0;

customer_buf[i].c_payment_cnt = 1;
customer_buf[i].c_delivery_cnt = 0;

// Generate CUSTOMER and HISTORY data

customer_buf[i].c_id = c[i].c_id;

strcpy(customer_buf[i].c_first, c[i].c_first);
strcpy(customer_buf[i].c_last, c[i].c_last);

customer_buf[i].c_middle[0] = 'O';
customer_buf[i].c_middle[1] = 'E';

MakeAddress(customer_buf[i].c_street_1,
            customer_buf[i].c_street_2,
            customer_buf[i].c_city,
            customer_buf[i].c_state,
            customer_buf[i].c_zip);

MakeNumberString(16, 16, PHONE_LEN,
customer_buf[i].c_phone);

if (RandomNumber(1L, 100L) > 10)
    customer_buf[i].c_credit[0] = 'G';
else
    customer_buf[i].c_credit[0] = 'B';
customer_buf[i].c_credit[1] = 'C';

customer_buf[i].c_credit_lim = 50000.0;
customer_buf[i].c_discount = ((float)
RandomNumber(0L, 5000L)) / 10000.0;

// fix to avoid ODBC float to numeric conversion
problem.

// customer_buf[i].c_balance = -10.0;
strcpy(customer_buf[i].c_balance,"-10.0");

MakeAlphaString(300, 500, C_DATA_LEN,
customer_buf[i].c_data);

// Generate HISTORY data
MakeAlphaString(12, 24, H_DATA_LEN,
customer_buf[i].h_data);

}

//=====
//
// Function : LoadCustomerTable
//
//=====

void LoadCustomerTable(LOADER_TIME_STRUCT *customer_time_start)
{
    int i;
    long c_id;
    short c_d_id;
    short c_w_id;
    char c_first[FIRST_NAME_LEN+1];
    char c_middle[MIDDLE_NAME_LEN+1];
    char c_last[LAST_NAME_LEN+1];

```

<pre> char c_street_1[ADDRESS_LEN+1]; char c_street_2[ADDRESS_LEN+1]; char c_city[ADDRESS_LEN+1]; char c_state[STATE_LEN+1]; char c_zip[ZIP_LEN+1]; char c_phone[PHONE_LEN+1]; char c_credit[CREDIT_LEN+1]; double c_credit_lim; double c_discount; // fix to avoid ODBC float to numeric conversion problem. // double c_balance; char c_balance[6]; double c_ytd_payment; short c_payment_cnt; short c_delivery_cnt; char c_data[C_DATA_LEN+1]; char c_since[C_SINCE_LEN+1]; RETCODE rc; rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0, FIRST_NAME_LEN, NULL, 0, 0, 4); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0, MIDDLE_NAME_LEN, NULL, 0, 0, 5); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN, NULL, 0, 0, 6); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN, NULL, 0, 0, 7); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN, NULL, 0, 0, 8); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL, 0, 0, 9); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0, 0, 10); if (rc != SUCCEED) </pre>	<pre> HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0, 11); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL, 0, 0, 12); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0, C_SINCE_LEN, NULL, 0, SQLCHARACTER, 13); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL, 0, 0, 14); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 15); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 16); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); // fix to avoid ODBC float to numeric conversion problem. // rc = bcp_bind(c_hdbc1, (BYTE *) &c_balance, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 17); // if (rc != SUCCEED) // HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0, SQLCHARACTER, 17); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 18); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 19); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 20); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0, 500, NULL, 0, 0, 21); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); for (i = 0; i < customers_per_district; i++) { c_id = customer_buf[i].c_id; c_d_id = customer_buf[i].c_d_id; c_w_id = customer_buf[i].c_w_id; </pre>
---	---

```

strcpy(c_first, customer_buf[i].c_first);
strcpy(c_middle, customer_buf[i].c_middle);
strcpy(c_last, customer_buf[i].c_last);
strcpy(c_street_1, customer_buf[i].c_street_1);
strcpy(c_street_2, customer_buf[i].c_street_2);
strcpy(c_city, customer_buf[i].c_city);
strcpy(c_state, customer_buf[i].c_state);
strcpy(c_zip, customer_buf[i].c_zip);
strcpy(c_phone, customer_buf[i].c_phone);
strcpy(c_credit, customer_buf[i].c_credit);

FormatDate(&c_since);

c_credit_lim = customer_buf[i].c_credit_lim;
c_discount = customer_buf[i].c_discount;

// fix to avoid ODBC float to numeric conversion
// c_balance = customer_buf[i].c_balance;
strcpy(c_balance, customer_buf[i].c_balance);

c_ytd_payment = customer_buf[i].c_ytd_payment;
c_payment_cnt = customer_buf[i].c_payment_cnt;
c_delivery_cnt = customer_buf[i].c_delivery_cnt;

strcpy(c_data, customer_buf[i].c_data);

// Send data to server
rc = bcp_sendrow(c_hdbc1);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

customer_rows_loaded++;
CheckForCommit(c_hdbc1, c_hstmt1,
customer_rows_loaded, "customer", &customer_time_start->time_start);
}
}

```

```

//=====
//
// Function : LoadHistoryTable
//
//=====

void LoadHistoryTable(LOADER_TIME_STRUCT *history_time_start)
{
    int i;
    long c_id;
    short c_d_id;
    short c_w_id;
    double h_amount;
    char h_data[H_DATA_LEN+1];
    char h_date[H_DATE_LEN+1];
    RETCODE rc;

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 2);

```

```

    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 3);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 4);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 5);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &h_date, 0, H_DATE_LEN,
NULL, 0, SQLCHARACTER, 6);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 7);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0, H_DATA_LEN, NULL, 0, 0,
8);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    for (i = 0; i < customers_per_district; i++)
    {
        c_id = customer_buf[i].c_id;
        c_d_id = customer_buf[i].c_d_id;
        c_w_id = customer_buf[i].c_w_id;
        h_amount = customer_buf[i].h_amount;
        strcpy(h_data, customer_buf[i].h_data);

        FormatDate(&h_date);

        // send to server
        rc = bcp_sendrow(c_hdbc2);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);

        history_rows_loaded++;
        CheckForCommit(c_hdbc2, c_hstmt2,
history_rows_loaded, "history", &history_time_start->time_start);
    }
}

//=====
//
// Function : LoadOrders
//
//=====

void LoadOrders()
{
    LOADER_TIME_STRUCT orders_time_start;
    LOADER_TIME_STRUCT new_order_time_start;

```

```

        LOADER_TIME_STRUCT    order_line_time_start;
        short                 w_id;
        short                 d_id;
        DWORD
dwThreadID[MAX_ORDER_THREADS];
        HANDLE
hThread[MAX_ORDER_THREADS];
        char                 name[20];
        RETCODE
rc;
        char
bcpHint[128];

        // seed with unique number
        seed(6);

        printf("Loading orders...\n");

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                BuildIndex("idxordcl");
                BuildIndex("idxnodel");
                BuildIndex("idxodlcl");
        }

        // initialize bulk copy
        sprintf(name, "%s..%s", aptr->database, "orders");

        rc = bcp_init(o_hdbc1, name, NULL, "logs\\orders.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcpHint, "tablock, order (o_w_id, o_d_id, o_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
                rc = bcp_control(o_hdbc1, BCPHINTS, (void*) bcpHint);

                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);
        }

        sprintf(name, "%s..%s", aptr->database, "new_order");

        rc = bcp_init(o_hdbc2, name, NULL, "logs\\neword.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcpHint, "tablock, order (no_w_id, no_d_id,
no_o_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 9000));
                rc = bcp_control(o_hdbc2, BCPHINTS, (void*) bcpHint);

                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc2);
        }

        sprintf(name, "%s..%s", aptr->database, "order_line");

        rc = bcp_init(o_hdbc3, name, NULL, "logs\\ordline.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))

```

```

        {
                sprintf(bcpHint, "tablock, order (ol_w_id, ol_d_id,
ol_o_id, ol_number), ROWS_PER_BATCH = %u", (aptr->num_warehouses *
300000));
                rc = bcp_control(o_hdbc3, BCPHINTS, (void*) bcpHint);

                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
        }

        orders_rows_loaded = 0;
        new_order_rows_loaded = 0;
        order_line_rows_loaded = 0;

        OrdersBufInit();

        orders_time_start.time_start = (TimeNow() / MILLI);
        new_order_time_start.time_start = (TimeNow() / MILLI);
        order_line_time_start.time_start = (TimeNow() / MILLI);

        for (w_id = (short)aptr->starting_warehouse; w_id <=
aptr->num_warehouses; w_id++)
        {
                for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE;
d_id++)
                {
                        OrdersBufLoad(d_id, w_id);

                        // start parallel loading threads here...

                        // start Orders table thread

                        printf("...Loading Order Table for: d_id = %d,
w_id = %d\n", d_id, w_id);

                        hThread[0] = CreateThread(NULL,

                                0,

                                (LPTHREAD_START_ROUTINE) LoadOrdersTable,

                                &orders_time_start,

                                0,

                                &dwThreadID[0]);

                        if (hThread[0] == NULL)
                        {
                                printf("Error, failed in creating
creating thread = 0.\n");
                                exit(-1);
                        }

                        // start NewOrder table thread

                        printf("...Loading New-Order Table for: d_id
= %d, w_id = %d\n", d_id, w_id);

                        hThread[1] = CreateThread(NULL,

                                0,

                                (LPTHREAD_START_ROUTINE) LoadNewOrderTable,

                                &new_order_time_start,

```

```

0,
&dwThreadId[1]);

    if (hThread[1] == NULL)
    {
        printf("Error, failed in creating
creating thread = 1.\n");
        exit(-1);
    }

    // start Order-Line table thread

    printf("...Loading Order-Line Table for: d_id
= %d, w_id = %d\n", d_id, w_id);

    hThread[2] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadOrderLineTable,
&order_line_time_start,
0,
&dwThreadId[2]);

    if (hThread[2] == NULL)
    {
        printf("Error, failed in creating
creating thread = 2.\n");
        exit(-1);
    }

    WaitForSingleObject( hThread[0], INFINITE
);
    WaitForSingleObject( hThread[1], INFINITE
);
    WaitForSingleObject( hThread[2], INFINITE
);

    if (CloseHandle(hThread[0]) == FALSE)
    {
        printf("Error, failed in closing
Orders thread handle with errno: %d\n", GetLastError());
    }

    if (CloseHandle(hThread[1]) == FALSE)
    {
        printf("Error, failed in closing
NewOrder thread handle with errno: %d\n", GetLastError());
    }

    if (CloseHandle(hThread[2]) == FALSE)
    {
        printf("Error, failed in closing
OrderLine thread handle with errno: %d\n", GetLastError());
    }
}

printf("Finished loading orders.\n");

return;

```

```

}

//=====
//
// Function : OrdersBufInit
//
// Clears shared buffer for ORDERS, NEWORDER, and ORDERLINE
//
//=====

void OrdersBufInit()
{
    int i;
    int j;

    for (i=0;i<orders_per_district;i++)
    {
        orders_buf[i].o_id = 0;
        orders_buf[i].o_d_id = 0;
        orders_buf[i].o_w_id = 0;
        orders_buf[i].o_c_id = 0;
        orders_buf[i].o_carrier_id = 0;
        orders_buf[i].o_ol_cnt = 0;
        orders_buf[i].o_all_local = 0;

        for (j=0;j<=14;j++)
        {
            orders_buf[i].o_ol[j].ol = 0;
            orders_buf[i].o_ol[j].ol_i_id = 0;

            orders_buf[i].o_ol[j].ol_supply_w_id = 0;
            orders_buf[i].o_ol[j].ol_quantity = 0;
            orders_buf[i].o_ol[j].ol_amount = 0;
            strcpy(orders_buf[i].o_ol[j].ol_dist_info,"");
        }
    }
}

//=====
//
// Function : OrdersBufLoad
//
// Fills shared buffer for ORDERS, NEWORDER, and ORDERLINE
//
//=====

void OrdersBufLoad(int d_id, int w_id)
{
    int cust[ORDERS_PER_DISTRICT+1];
    long o_id;
    short ol;

    printf("...Loading Order Buffer for: d_id = %d, w_id = %d\n",
d_id, w_id);

    GetPermutation(cust, orders_per_district);

    for (o_id=0;o_id<orders_per_district;o_id++)
    {

```

```

// Generate ORDER and NEW-ORDER data

orders_buf[o_id].o_d_id = d_id;
orders_buf[o_id].o_w_id = w_id;
orders_buf[o_id].o_id = o_id+1;
orders_buf[o_id].o_c_id = cust[o_id+1];
orders_buf[o_id].o_ol_cnt = (short)RandomNumber(5L,
15L);

if (o_id < first_new_order)
{
    orders_buf[o_id].o_carrier_id =
(short)RandomNumber(1L, 10L);
    orders_buf[o_id].o_all_local = 1;
}
else
{
    orders_buf[o_id].o_carrier_id = 0;
    orders_buf[o_id].o_all_local = 1;
}

for (ol=0; ol<orders_buf[o_id].o_ol_cnt; ol++)
{
    orders_buf[o_id].o_ol[ol].ol = ol+1;
    orders_buf[o_id].o_ol[ol].ol_i_id =
RandomNumber(1L, max_items);
    orders_buf[o_id].o_ol[ol].ol_supply_w_id =
w_id;
    orders_buf[o_id].o_ol[ol].ol_quantity = 5;
    MakeAlphaString(24, 24,
OL_DIST_INFO_LEN, &orders_buf[o_id].o_ol[ol].ol_dist_info);

    // Generate ORDER-LINE data
    if (o_id < first_new_order)
    {
        orders_buf[o_id].o_ol[ol].ol_amount = 0;

        // Added to insure ol_delivery_d
        set properly during load

        FormatDate(&orders_buf[o_id].o_ol[ol].ol_delivery_d);

        }
        else
        {
            orders_buf[o_id].o_ol[ol].ol_amount = RandomNumber(1,999999)/100.0;

            // Added to insure ol_delivery_d
            set properly during load

            // odbc datetime format
            strcpy(orders_buf[o_id].o_ol[ol].ol_delivery_d,"1899-12-31 00:00:00.000");

        }
    }
}

//=====
//

```

```

// Function : LoadOrdersTable
//
//=====

void LoadOrdersTable(LOADER_TIME_STRUCT *orders_time_start)
{
    int i;
    long o_id;
    short o_d_id;
    short o_w_id;
    long o_c_id;
    short o_carrier_id;
    short o_ol_cnt;
    short o_all_local;
    char o_entry_d[O_ENTRY_D_LEN+1];
    RETCODE rc;
    DBINT rcint;

    // bind ORDER data
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 2);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 3);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT4, 4);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_entry_d, 0,
O_ENTRY_D_LEN, NULL, 0, SQLCHARACTER, 5);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 6);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 7);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 8);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    for (i = 0; i < orders_per_district; i++)
    {
        o_id = orders_buf[i].o_id;
        o_d_id = orders_buf[i].o_d_id;
        o_w_id = orders_buf[i].o_w_id;
        o_c_id = orders_buf[i].o_c_id;
        o_carrier_id = orders_buf[i].o_carrier_id;
        o_ol_cnt = orders_buf[i].o_ol_cnt;
    }
}

```

```

o_all_local = orders_buf[i].o_all_local;

FormatDate(&o_entry_d);

// send data to server
rc = bcp_sendrow(o_hdbc1);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc1);

orders_rows_loaded++;
CheckForCommit(o_hdbc1, o_hstmt1,
orders_rows_loaded, "orders", &orders_time_start->time_start);
}

// rcint = bcp_batch(o_hdbc1);
// if (rcint < 0)
//     HandleErrorDBC(o_hdbc1);

if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
{
    rcint = bcp_done(o_hdbc1);
    if (rcint < 0)
        HandleErrorDBC(o_hdbc1);

    SQLFreeStmt(o_hstmt1, SQL_DROP);
    SQLDisconnect(o_hdbc1);
    SQLFreeConnect(o_hdbc1);

    // if build index after load...
    if ((aptr->build_index == 1) && (aptr->index_order ==
0))

        BuildIndex("idxordcl");

    // build non-clustered index
    if (aptr->build_index == 1)
        BuildIndex("idxordnc");
}

}

//=====
//
// Function : LoadNewOrderTable
//
//=====

void LoadNewOrderTable(LOADER_TIME_STRUCT *new_order_time_start)
{
    int i;
    long o_id;
    short o_d_id;
    short o_w_id;
    RETCODE rc;
    DBINT rcint;

    // Bind NEW-ORDER data

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 2);
    if (rc != SUCCEED)

```

```

HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 3);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

    for (i = first_new_order; i < last_new_order; i++)
    {
        o_id = orders_buf[i].o_id;
        o_d_id = orders_buf[i].o_d_id;
        o_w_id = orders_buf[i].o_w_id;

        rc = bcp_sendrow(o_hdbc2);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc2);

        new_order_rows_loaded++;
        CheckForCommit(o_hdbc2, o_hstmt2,
new_order_rows_loaded, "new_order", &new_order_time_start->time_start);
    }

    // rcint = bcp_batch(o_hdbc2);
    // if (rcint < 0)
    //     HandleErrorDBC(o_hdbc2);

    if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
    {
        rcint = bcp_done(o_hdbc2);
        if (rcint < 0)
            HandleErrorDBC(o_hdbc2);

        SQLFreeStmt(o_hstmt2, SQL_DROP);
        SQLDisconnect(o_hdbc2);
        SQLFreeConnect(o_hdbc2);

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order ==
0))

            BuildIndex("idxnordcl");

    }

}

//=====
//
// Function : LoadOrderLineTable
//
//=====

void LoadOrderLineTable(LOADER_TIME_STRUCT *order_line_time_start)
{
    int i,j;
    long o_id;
    short o_d_id;
    short o_w_id;
    long ol;
    long ol_i_id;
    short ol_supply_w_id;
    short ol_quantity;
    double ol_amount;
    char ol_dist_info[DIST_INFO_LEN+1];
    char ol_delivery_d[OL_DELIVERY_D_LEN+1];
    RETCODE rc;

```

```

        DBINT          rcint;

        // bind ORDER-LINE data
        rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0, SQL_VARLEN_DATA,
        NULL, 0, SQLINT4, 1);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc3);

        rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA,
        NULL, 0, SQLINT2, 2);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc3);

        rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA,
        NULL, 0, SQLINT2, 3);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc3);

        rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0, SQL_VARLEN_DATA, NULL,
        0, SQLINT4, 4);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc3);

        rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0, SQL_VARLEN_DATA,
        NULL, 0, SQLINT4, 5);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc3);

        rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id, 0,
        SQL_VARLEN_DATA, NULL, 0, SQLINT2, 6);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc3);

        rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0,
        OL_DELIVERY_D_LEN, NULL, 0, SQLCHARACTER, 7);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc3);

        rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0, SQL_VARLEN_DATA,
        NULL, 0, SQLINT2, 8);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc3);

        rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0, SQL_VARLEN_DATA,
        NULL, 0, SQLFLT8, 9);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc3);

        rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0, DIST_INFO_LEN,
        NULL, 0, 0, 10);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc3);

        for (i = 0; i < orders_per_district; i++)
        {
            o_id = orders_buf[i].o_id;
            o_d_id = orders_buf[i].o_d_id;
            o_w_id = orders_buf[i].o_w_id;

            for (j=0; j < orders_buf[i].o_ol_cnt; j++)
            {
                ol = orders_buf[i].o_ol[j].ol;
                ol_i_id = orders_buf[i].o_ol[j].ol_i_id;
                ol_supply_w_id =
orders_buf[i].o_ol[j].ol_supply_w_id;
                ol_quantity =
orders_buf[i].o_ol[j].ol_quantity;

```

```

                ol_amount =
orders_buf[i].o_ol[j].ol_amount;

                strcpy(ol_delivery_d, orders_buf[i].o_ol[j].ol_delivery_d);

                strcpy(ol_dist_info, orders_buf[i].o_ol[j].ol_dist_info);

                rc = bcp_sendrow(o_hdbc3);
                if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc3);

                order_line_rows_loaded++;
                CheckForCommit(o_hdbc3, o_hstmt3,
                order_line_rows_loaded, "order_line", &order_line_time_start->time_start);
            }
        }

        // rcint = bcp_batch(o_hdbc3);
        // if (rcint < 0)
        //     HandleErrorDBC(o_hdbc3);

        if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
        {
            rcint = bcp_done(o_hdbc3);
            if (rcint < 0)
                HandleErrorDBC(o_hdbc3);

            SQLFreeStmt(o_hstmt3, SQL_DROP);
            SQLDisconnect(o_hdbc3);
            SQLFreeConnect(o_hdbc3);

            // if build index after load...
            if ((aptr->build_index == 1) && (aptr->index_order ==
0))
                BuildIndex("idxodlcl");
        }
    }

//=====
//
// Function : GetPermutation
//
//=====

void GetPermutation(int perm[], int n)
{
    int i, r, t;

    for (i=1; i<=n; i++)
        perm[i] = i;

    for (i=1; i<=n; i++)
    {
        r = RandomNumber(i,n);
        t = perm[i];
        perm[i] = perm[r];
        perm[r] = t;
    }
}

```

```

//=====
//
// Function : CheckForCommit
//
//=====

void CheckForCommit(HDBC hdbc,
                    HSTMT hstmt,
                    int rows_loaded,
                    char *table_name,
                    long *time_start)
{
    long    time_end, time_diff;
    // DBINT rcint;

    if ( !(rows_loaded % aptr->batch) )
    {
        // rcint = bcp_batch(hdbc);
        // if (rcint < 0)
        //     HandleErrorDBC(hdbc);

        time_end = (TimeNow() / MILLI);
        time_diff = time_end - *time_start;

        printf("> Loaded %ld rows into %s in %ld sec - Total =
%d (%.2f rps)\n",
                aptr->batch,
                table_name,
                time_diff,
                rows_loaded,
                (float) aptr->batch / (time_diff ?
time_diff : 1L));

        *time_start = time_end;
    }

    return;
}

//=====
//
// Function : OpenConnections
//
//=====

void OpenConnections()
{
    RETCODE    rc;

    char        szDriverString[300];
    char        szDriverStringOut[1024];
    SQLSMALLINT cbDriverStringOut;

    SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE,
&henv );

    SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION,
(void*)SQL_OV_ODBC3, 0 );

```

```

SQLAllocHandle(SQL_HANDLE_DBC, henv , &i_hdbc1);
SQLAllocHandle(SQL_HANDLE_DBC, henv , &w_hdbc1);
SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc1);
SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc2);
SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc1);
SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc2);
SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc3);

SQLSetConnectAttr(i_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
SQLSetConnectAttr(w_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
SQLSetConnectAttr(c_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
SQLSetConnectAttr(c_hdbc2, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );

// Open connections to SQL Server

// Connection 1

sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
                                aptr->user,
aptr->password,
aptr->database );

rc = SQLSetConnectOption (i_hdbc1, SQL_PACKET_SIZE,
aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(i_hdbc1);

rc = SQLDriverConnect ( i_hdbc1,
                                NULL,
(SQLCHAR*)&szDriverString[0] ,
                                SQL_NTS,
(SQLCHAR*)&szDriverStringOut[0],
sizeof(szDriverStringOut),
&cbDriverStringOut,
SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
    HandleErrorDBC(i_hdbc1);

// Connection 2

sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
                                aptr->user,
aptr->password,

```

<pre> aptr->database); rc = SQLSetConnectOption (w_hdbc1, SQL_PACKET_SIZE, aptr->pack_size); if (rc != SUCCEED) HandleErrorDBC(w_hdbc1); rc = SQLDriverConnect (w_hdbc1, NULL, (SQLCHAR*)&szDriverString[0] , SQL_NTS, (SQLCHAR*)&szDriverStringOut[0], sizeof(szDriverStringOut), &cbDriverStringOut, SQL_DRIVER_NOPROMPT); if (rc != SUCCEED) HandleErrorDBC(w_hdbc1); // Connection 3 sprintf(szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" , aptr->server, aptr->password, aptr->database); rc = SQLSetConnectOption (c_hdbc1, SQL_PACKET_SIZE, aptr->pack_size); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); rc = SQLDriverConnect (c_hdbc1, NULL, (SQLCHAR*)&szDriverString[0] , SQL_NTS, (SQLCHAR*)&szDriverStringOut[0], sizeof(szDriverStringOut), &cbDriverStringOut, SQL_DRIVER_NOPROMPT); if (rc != SUCCEED) HandleErrorDBC(c_hdbc1); // Connection 4 sprintf(szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" , aptr->server, </pre>	<pre> aptr->password, aptr->database); rc = SQLSetConnectOption (c_hdbc2, SQL_PACKET_SIZE, aptr->pack_size); if (rc != SUCCEED) HandleErrorDBC(c_hdbc2); rc = SQLDriverConnect (c_hdbc2, NULL, (SQLCHAR*)&szDriverString[0] , SQL_NTS, (SQLCHAR*)&szDriverStringOut[0], sizeof(szDriverStringOut), &cbDriverStringOut, SQL_DRIVER_NOPROMPT); if (rc != SUCCEED) HandleErrorDBC(c_hdbc2); // Connection 5 sprintf(szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" , aptr->server, aptr->password, aptr->database); rc = SQLSetConnectOption (o_hdbc1, SQL_PACKET_SIZE, aptr->pack_size); if (rc != SUCCEED) HandleErrorDBC(o_hdbc1); rc = SQLDriverConnect (o_hdbc1, NULL, (SQLCHAR*)&szDriverString[0] , SQL_NTS, (SQLCHAR*)&szDriverStringOut[0], sizeof(szDriverStringOut), &cbDriverStringOut, SQL_DRIVER_NOPROMPT); if (rc != SUCCEED) HandleErrorDBC(o_hdbc1); // Connection 6 sprintf(szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" , aptr->server, </pre>
---	---

```

aptr->server,

aptr->password,

aptr->database );

        rc = SQLSetConnectOption (o_hdbc2, SQL_PACKET_SIZE,
aptr->pack_size);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc2);

        rc = SQLDriverConnect ( o_hdbc2,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc2);

        // Connection 7

        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,

aptr->password,

aptr->database );

        rc = SQLSetConnectOption (o_hdbc3, SQL_PACKET_SIZE,
aptr->pack_size);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc3);

        rc = SQLDriverConnect ( o_hdbc3,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc3);
    }

```

aptr->user,

aptr->user,

```

//=====
//
// Function name: BuildIndex
//
//=====

void BuildIndex(char *index_script)
{
    char    cmd[256];

    printf("Starting index creation: %s\n",index_script);

    sprintf(cmd, "isql -S%s -U%s -P%s -e -i%s\\%s.sql > logs\\%s.log",
aptr->server,
aptr->user,
aptr->password,
aptr->index_script_path,
index_script,
index_script);

    system(cmd);

    printf("Finished index creation: %s\n",index_script);
}

void HandleErrorDBC (SQLHDBC hdbc1)
{
    SQLCHAR                SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
    SQLINTEGER NativeError;
    SQLSMALLINT i, MsgLen;
    SQLRETURN rc2;
    char                timebuf[128];
    char                datebuf[128];
    FILE                *fp1;

    i = 1;
    while (( rc2 = SQLGetDiagRec(SQL_HANDLE_DBC , hdbc1, i,
SqlState , &NativeError,
Msg, sizeof(Msg) , &MsgLen ))
!= SQL_NO_DATA )
    {

        sprintf( szLastError , "%s" , Msg );

        _strtime(timebuf);
        _strdate(datebuf);

        printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

        fp1 = fopen("logs\\tpccldr.err","w");
        if (fp1 == NULL)
            printf("ERROR: Unable to open errorlog
file.\n");

        else
        {
            fprintf(fp1, "[%s : %s] %s\n" , datebuf,
timebuf, szLastError);
            fclose(fp1);
        }

        i++;
    }
}

```

```

}

void HandleErrorSTMT (HSTMT hstmt1)
{
    SQLCHAR          SqlState[6],
    Msg[SQL_MAX_MESSAGE_LENGTH];
    SQLINTEGER NativeError;
    SQLSMALLINT i, MsgLen;
    SQLRETURN rc2;
    char             timebuf[128];
    char             datebuf[128];
    FILE             *fp1;

    i = 1;
    while (( rc2 = SQLGetDiagRec(SQL_HANDLE_STMT , hstmt1, i,
    SqlState , &NativeError,
                                Msg, sizeof(Msg) , &MsgLen )
    != SQL_NO_DATA )
    {

        sprintf( szLastError , "%s" , Msg );

        _strtime(timebuf);
        _strdate(datebuf);

        printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

        fp1 = fopen("logs\\tpccldr.err", "w");
        if (fp1 == NULL)
            printf("ERROR: Unable to open errorlog
file.\n");
        else
        {
            fprintf(fp1, "[%s : %s] %s\n" , datebuf,
timebuf, szLastError);
            fclose(fp1);
        }

        i++;
    }
}

void FormatDate ( char* szTimeCOutput )
{
    struct tm when;
    time_t now;

    time( &now );
    when = *localtime( &now );

    mktime( &when );

    // odbc datetime format
    strftime( szTimeCOutput , 30 , "%Y-%m-%d %H:%M:%S.000",
    &when );

    return;
}

//=====
//
// Function : CheckSQL

```

```

//=====
//=====

void CheckSQL()
{
    RETCODE          rc;

    char             szDriverString[300];
    char             szDriverStringOut[1024];
    int              SQLBuildFlag;
    char             resp;

    SQLSMALLINT      cbDriverStringOut;
    SQLCHAR          SQLVersion[19];
    SQLINTEGER       SQLVersionInd;

    SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE,
    &henv );

    SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION,
    (void*)SQL_OV_ODBC3, 0 );

    SQLAllocHandle(SQL_HANDLE_DBC, henv , &v_hdbc);

    SQLSetConnectAttr(v_hdbc, SQL_COPT_SS_BCP, (void
    *)SQL_BCP_ON, SQL_IS_INTEGER );

    // Open connection to SQL Server

    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s" ,
    aptr->server,
                                aptr->user,
    aptr->password );

    if ( SQLSetConnectAttr( v_hdbc, SQL_ATTR_PACKET_SIZE,
    (SQLPOINTER)aptr->pack_size, SQL_IS_INTEGER ) != SQL_SUCCESS )
        HandleErrorDBC(v_hdbc);

    rc = SQLDriverConnect ( v_hdbc,
                                NULL,
    (SQLCHAR*)&szDriverString[0] ,
                                SQL_NTS,
    (SQLCHAR*)&szDriverStringOut[0],
    sizeof(szDriverStringOut),
    &cbDriverStringOut,
    SQL_DRIVER_NOPROMPT );

    if ((rc != SQL_SUCCESS) && (rc !=
    SQL_SUCCESS_WITH_INFO))
        HandleErrorDBC(v_hdbc);

    if ( SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt)
    != SQL_SUCCESS )
        HandleErrorSTMT(v_hstmt);

```

```

        rc = SQLBindCol(v_hstmt, 4, SQL_C_CHAR, &SQLVersion,
sizeof(SQLVersion), &SQLVersionInd);

```

```

        // issue SQL Server extended stored procedure (xp_msver) to
determine installed version

```

```

        rc = SQLExecDirect(v_hstmt, "EXECUTE xp_msver
ProductVersion", SQL_NTS);

```

```

        if ((rc != SQL_SUCCESS) && (rc !=
SQL_SUCCESS_WITH_INFO))
            HandleErrorSTMT(v_hstmt);

```

```

        rc = SQLFetch(v_hstmt);

```

```

        if (rc != SQL_SUCCESS)
            HandleErrorDBC(v_hdbc);

```

```

        // Check build number to ensure 8.00.194 or higher

```

```

        SQLBuildFlag = 1;

```

```

        // first check the Major version

```

```

        if ( SQLVersion[0] == '8' )
        {
            if (( SQLVersion[2] == '0' ) & ( SQLVersion[3] == '0' ) )
            {

```

```

                if ( SQLVersion[5] == '1' )
                {

```

```

                    if ( (SQLVersion[6] == '9') &

```

```

(SQLVersion[7] == '4') )

```

```

                    {
                        SQLBuildFlag = 0;
                        printf("You are using

```

```

SQL Server version = %9s\n\n", SQLVersion);

```

```

                    }

```

```

                    else

```

```

                    {
                        SQLBuildFlag = 1;

```

```

                    }

```

```

                }

```

```

                else

```

```

                {

```

```

                    if ( SQLVersion[5] == '3' )

```

```

                    {
                        if ( (SQLVersion[6] >=

```

```

53) & (SQLVersion[7] >= 48) )

```

```

                    {

```

```

SQLBuildFlag = 0;

```

```

are using SQL Server version = %9s\n\n", SQLVersion);

```

```

                    }

```

```

                    else

```

```

                    {

```

```

SQLBuildFlag = 1;

```

```

                    }

```

```

                }

```

```

            }

```

```

        }

```

```

    }
    else
    {

```

```

        SQLBuildFlag = 1;

```

```

    }

```

```

    if ( SQLBuildFlag == 1 )

```

```

    {
        printf("NOTE: The SQL Server version you are using is
not supported\n");

```

```

        printf("for TPC-C benchmarking. You currently have
SQL Server version %9s\n",SQLVersion);
        printf("installed. Please upgrade to Microsoft SQL
Server 2000 (8.00.0194) or better.\n");

```

```

        printf("and re-run the SETUP program.\n\n");
        printf("Do you wish to continue with setup? (Y/N): ");
        resp = getchar();
        if ( ( resp == 'N' ) || (resp == 'n') )
        {

```

```

            printf("\nSetup Aborted!\n");
            exit(1);

```

```

        }

```

```

        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);

```

```

        SQLDisconnect(v_hdbc);

```

```

        SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

```

```

        return;

```

```

    }

```

```

//=====

```

```

//
// Function : CheckDataBase
//
//=====

```

```

void CheckDataBase()

```

```

{

```

```

    RETCODE rc;

```

```

    char szDriverString[300];

```

```

    char szDriverStringOut[1024];

```

```

    char TablesBitMap[9] =

```

```

{"0000000000"};

```

```

    int i, ExitFlag;

```

```

    SQLSMALLINT cbDriverStringOut;

```

```

    SQLCHAR TabName[10];

```

```

    SQLINTEGER TabNameInd, TabCount,

```

```

    TabCountInd;

```

```

    ExitFlag = 0;

```

```

    SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE,
&henv );

```

```

    SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION,
(void*)SQL_OV_ODBC3, 0 );

```

```

    SQLAllocHandle(SQL_HANDLE_DBC, henv , &v_hdbc);

```

```

    SQLSetConnectAttr(v_hdbc, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );

```

```

    // Open connection to SQL Server

```

```

    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

```

```

aptr->server,
aptr->password,
aptr->database );

rc = SQLSetConnectAttr( v_hdbc, SQL_ATTR_PACKET_SIZE,
(SQLPOINTER)aptr->pack_size, SQL_IS_UINTEGER );
if (rc != SQL_SUCCESS)
    HandleErrorDBC(v_hdbc);

rc = SQLDriverConnect ( v_hdbc,
                        NULL,
(SQLCHAR*)&szDriverString[0] ,
                        SQL_NTS,
(SQLCHAR*)&szDriverStringOut[0],
sizeof(szDriverStringOut),
&cbDriverStringOut,
SQL_DRIVER_NOPROMPT );

// if the rc is SQL_ERROR, the the TPCC database probably does
not exist
if (rc == SQL_ERROR)
{
    printf("The database TPCC does not appear to exist!\n");
    printf("\nCheck LOGS\\ directory for database creation
errors.\n");

    // cleanup database connections and handles
    SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
    SQLDisconnect(v_hdbc);
    SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

    // since there is not a database, exit back to SETUP.CMD
    exit(1);
}

if ( SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt)
!= SQL_SUCCESS )
    HandleErrorDBC(v_hdbc);

if ( SQLBindCol(v_hstmt, 1, SQL_C_ULONG, &TabCount, 0,
&TabCountInd) != SQL_SUCCESS )
    HandleErrorSTMT(v_hstmt);

// count the number of user tables from sysobjects
rc = SQLExecDirect(v_hstmt, "select count(*) from sysobjects
where xtype = 'U'", SQL_NTS);
if ((rc != SQL_SUCCESS) && (rc !=
SQL_SUCCESS_WITH_INFO))
    HandleErrorSTMT(v_hstmt);

if ( SQLFetch(v_hstmt) != SQL_SUCCESS )
    HandleErrorSTMT(v_hstmt);

// if the number of tables is less than 9, select all the user tables in
TPCC
if (TabCount != 9)
{
    SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);

```

```

SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc ,
&v_hstmt);

if ( SQLBindCol(v_hstmt, 1, SQL_C_CHAR,
&TabName, sizeof(TabName), &TabNameInd) != SQL_SUCCESS )
    HandleErrorSTMT(v_hstmt);

// select the list of user tables into a result set
rc = SQLExecDirect(v_hstmt, "select * from sysobjects
where xtype = 'U'", SQL_NTS);
if ((rc != SQL_SUCCESS) && (rc !=
SQL_SUCCESS_WITH_INFO))
    HandleErrorSTMT(v_hstmt);

// go through the result set and set the bitmap for each
found table
// set the bitmap to '1' if the table name is found

while ((rc = SQLFetch(v_hstmt)) != SQL_NO_DATA)
{
    switch( TabName[0] )
    {
        case 'w':
            TablesBitMap[0] = '1';
            break;
        case 'd':
            TablesBitMap[1] = '1';
            break;
        case 'c':
            TablesBitMap[2] = '1';
            break;
        case 'h':
            TablesBitMap[3] = '1';
            break;
        case 'n':
            TablesBitMap[4] = '1';
            break;
        case 'o':
            if (TabName[5] = 's')
                TablesBitMap[5] = '1';
            if (TabName[5] = '_')
                TablesBitMap[6] = '1';
            break;
        case 't':
            TablesBitMap[7] = '1';
            break;
        case 's':
            TablesBitMap[8] = '1';
            break;
    }
}

// a '0' ExitFlag means do NOT exit the loader early, a '1'
means exit the loader early
ExitFlag = 0;

// iterate through the bitmap to display which table(s) is
actually missing
for (i = 0; i <= 8; i++)
{
    switch(i)
    {
        case 0:
            if (TablesBitMap[i] == '0')
            {
                printf("The Warehouse
table is missing or damaged.\n");
                ExitFlag = 1;
            }
        }
    }
}

```

	<pre> } break; case 1: if (TablesBitMap[i] == '0') { printf("The District table is missing or damaged.\n"); ExitFlag = 1; } break; case 2: if (TablesBitMap[i] == '0') { printf("The Customer table is missing or damaged.\n"); ExitFlag = 1; } break; case 3: if (TablesBitMap[i] == '0') { printf("The History table is missing or damaged.\n"); ExitFlag = 1; } break; case 4: if (TablesBitMap[i] == '0') { printf("The New_Order table is missing or damaged.\n"); ExitFlag = 1; } break; case 5: if (TablesBitMap[i] == '0') { printf("The Orders table is missing or damaged.\n"); ExitFlag = 1; } break; case 6: if (TablesBitMap[i] == '0') { printf("The Order_Line table is missing or damaged.\n"); ExitFlag = 1; } break; case 7: if (TablesBitMap[i] == '0') { printf("The Item table is missing or damaged.\n"); ExitFlag = 1; } break; case 8: if (TablesBitMap[i] == '0') { printf("The Stock table is missing or damaged.\n"); ExitFlag = 1; } break; } } </pre>	<pre> // if one or more tables are missing, display message and exit the loader if (ExitFlag = 1) { printf("\nExiting TPC-C Loader!\n"); printf("\nCheck LOGS\\ directory for database\n"); printf("or table creation errors.\n"); // cleanup database connections and handles SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt); SQLDisconnect(v_hdbc); SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc); exit(1); } } // cleanup database connections and handles SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt); SQLDisconnect(v_hdbc); SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc); return; } </pre>
--	--	--

tpccldr.mak

```

# Microsoft Developer Studio Generated NMAKE File, Format Version 4.10
# ** DO NOT EDIT **

# TARGETTYPE "Win32 (x86) Console Application" 0x0103

!IF "$(CFG)" == ""
CFG=tpccldr - Win32 Debug
!MESSAGE No configuration specified. Defaulting to tpccldr - Win32 Debug.
!ENDIF

!IF "$(CFG)" != "tpccldr - Win32 Release" && "$(CFG)" !=\
"tpccldr - Win32 Debug"
!MESSAGE Invalid configuration "$(CFG)" specified.
!MESSAGE You can specify a configuration when running NMAKE on this
makefile
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tpccldr.mak" CFG="tpccldr - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpccldr - Win32 Release" (based on "Win32 (x86) Console
Application")
!MESSAGE "tpccldr - Win32 Debug" (based on "Win32 (x86) Console
Application")
!MESSAGE
!ERROR An invalid configuration is specified.
!ENDIF

!IF "$(OS)" == "Windows_NT"
NULL=
!ELSE
NULL=nul
!ENDIF
#####
#####

```

```

# Begin Project
# PROP Target_Last_Scanned "tpcldr - Win32 Debug"
RSC=rc.exe
CPP=cl.exe

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir "bin"
# PROP Intermediate_Dir "objects"
# PROP Target_Dir ""
OUTDIR=.bin
INTDIR=.objects

ALL : "$(OUTDIR)\tpcldr.exe"

CLEAN :
    -@erase "$(INTDIR)\getargs.obj"
    -@erase "$(INTDIR)\random.obj"
    -@erase "$(INTDIR)\strings.obj"
    -@erase "$(INTDIR)\time.obj"
    -@erase "$(INTDIR)\tpcldr.obj"
    -@erase "$(OUTDIR)\tpcldr.exe"

"$$(OUTDIR)" :
    if not exist "$(OUTDIR)/$(NULL)" mkdir "$(OUTDIR)"

"$$(INTDIR)" :
    if not exist "$(INTDIR)/$(NULL)" mkdir "$(INTDIR)"

# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
# "_CONSOLE" /YX /c
# ADD CPP /nologo /MT /W3 /GX /O2 /I "c:\mssql\dblib\include" /D
# "NDEBUG" /D "WIN32" /D "_CONSOLE" /D "DBNTWIN32" /c
# SUBTRACT CPP /YX
CPP_PROJ=/nologo /MT /W3 /GX /O2 /I "c:\mssql\dblib\include" /D
# "NDEBUG" /D\
# "WIN32" /D "_CONSOLE" /D "DBNTWIN32" /Fo"$$(INTDIR)/" /c
CPP_OBJS=.objects/
CPP_SBRS=.
# ADD BASE RSC /I 0x409 /d "NDEBUG"
# ADD RSC /I 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
BSC32_FLAGS=/nologo /o"$$(OUTDIR)\tpcldr.bsc"
BSC32_SBRS= \

LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib
odbc32.lib /nologo /subsystem:console /machine:I386
# ADD LINK32 c:\mssql\dblib\lib\ntwdblib.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbccp32.lib /nologo /subsystem:console /pdb:none
/machine:I386
LINK32_FLAGS=c:\mssql\dblib\lib\ntwdblib.lib kernel32.lib user32.lib
gdi32.lib\
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib\
uuid.lib odbccp32.lib /nologo /subsystem:console /pdb:none\
/machine:I386 /out:"$(OUTDIR)\tpcldr.exe"
LINK32_OBJS= \

```

```

"$$(INTDIR)\getargs.obj" \
"$$(INTDIR)\random.obj" \
"$$(INTDIR)\strings.obj" \
"$$(INTDIR)\time.obj" \
"$$(INTDIR)\tpcldr.obj"

"$$(OUTDIR)\tpcldr.exe" : "$(OUTDIR)" $(DEF_FILE) $(LINK32_OBJS)
    $(LINK32) @<<
    $(LINK32_FLAGS) $(LINK32_OBJS)
<<

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir "bin"
# PROP Intermediate_Dir "objects"
# PROP Target_Dir ""
OUTDIR=.bin
INTDIR=.objects

ALL : "$(OUTDIR)\tpcldr.exe"

CLEAN :
    -@erase "$(INTDIR)\getargs.obj"
    -@erase "$(INTDIR)\random.obj"
    -@erase "$(INTDIR)\strings.obj"
    -@erase "$(INTDIR)\time.obj"
    -@erase "$(INTDIR)\tpcldr.obj"
    -@erase "$(INTDIR)\vc40.idb"
    -@erase "$(INTDIR)\vc40.pdb"
    -@erase "$(OUTDIR)\tpcldr.exe"

"$$(OUTDIR)" :
    if not exist "$(OUTDIR)/$(NULL)" mkdir "$(OUTDIR)"

"$$(INTDIR)" :
    if not exist "$(INTDIR)/$(NULL)" mkdir "$(INTDIR)"

# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG"
# /D " _CONSOLE" /YX /c
# ADD CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /I "c:\mssql\dblib\include" /D
# "_DEBUG" /D "WIN32" /D "_CONSOLE" /D "DBNTWIN32" /c
# SUBTRACT CPP /YX
CPP_PROJ=/nologo /MTd /W3 /Gm /GX /Zi /Od /I "c:\mssql\dblib\include" /D\
# "_DEBUG" /D "WIN32" /D "_CONSOLE" /D "DBNTWIN32"
# /Fo"$$(INTDIR)/" \
# /Fd"$$(INTDIR)/" /c
CPP_OBJS=.objects/
CPP_SBRS=.
# ADD BASE RSC /I 0x409 /d "_DEBUG"
# ADD RSC /I 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
BSC32_FLAGS=/nologo /o"$$(OUTDIR)\tpcldr.bsc"
BSC32_SBRS= \

LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib
odbc32.lib /nologo /subsystem:console /debug /machine:I386

```

<pre># ADD LINK32 c:\mssql\dblib\lib\ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib /nologo /subsystem:console /pdb:none /debug /machine:I386 LINK32_FLAGS=c:\mssql\dblib\lib\ntwdblib.lib kernel32.lib user32.lib gdi32.lib\ winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib\ uuid.lib odbcc32.lib odbccp32.lib /nologo /subsystem:console /pdb:none /debug\ /machine:I386 /out:"\$(OUTDIR)\tpccldr.exe" LINK32_OBJS= \ "\$(INTDIR)\getargs.obj" \ "\$(INTDIR)\random.obj" \ "\$(INTDIR)\strings.obj" \ "\$(INTDIR)\time.obj" \ "\$(INTDIR)\tpccldr.obj" "\$(OUTDIR)\tpccldr.exe" : "\$(OUTDIR)" \$(DEF_FILE) \$(LINK32_OBJS) \$(LINK32) @<< \$(LINK32_FLAGS) \$(LINK32_OBJS) << !ENDIF .c{\$(CPP_OBJS)}.obj: \$(CPP) \$(CPP_PROJ) \$< .cpp{\$(CPP_OBJS)}.obj: \$(CPP) \$(CPP_PROJ) \$< .cxx{\$(CPP_OBJS)}.obj: \$(CPP) \$(CPP_PROJ) \$< .c{\$(CPP_SBRs)}.sbr: \$(CPP) \$(CPP_PROJ) \$< .cpp{\$(CPP_SBRs)}.sbr: \$(CPP) \$(CPP_PROJ) \$< .cxx{\$(CPP_SBRs)}.sbr: \$(CPP) \$(CPP_PROJ) \$< ##### ##### # Begin Target # Name "tpccldr - Win32 Release" # Name "tpccldr - Win32 Debug" !IF "\$(CFG)" == "tpccldr - Win32 Release" !ELSEIF "\$(CFG)" == "tpccldr - Win32 Debug" !ENDIF ##### ##### # Begin Source File SOURCE=. \src\random.c DEP_CPP_RAND0=\ ".\src\tpcc.h"\ "mssql\dblib\include\sqlldb.h"\ "mssql\dblib\include\sqlfront.h" "\$(INTDIR)\random.obj" : \$(SOURCE) \$(DEP_CPP_RAND0) "\$(INTDIR)" \$(CPP) \$(CPP_PROJ) \$(SOURCE) # End Source File # End Target # End Project</pre>	<pre># End Source File ##### # Begin Source File SOURCE=. \src\strings.c DEP_CPP_STRIN=\ ".\src\tpcc.h"\ "mssql\dblib\include\sqlldb.h"\ "mssql\dblib\include\sqlfront.h" "\$(INTDIR)\strings.obj" : \$(SOURCE) \$(DEP_CPP_STRIN) "\$(INTDIR)" \$(CPP) \$(CPP_PROJ) \$(SOURCE) # End Source File ##### # Begin Source File SOURCE=. \src\time.c DEP_CPP_TIME_=\ ".\src\tpcc.h"\ "mssql\dblib\include\sqlldb.h"\ "mssql\dblib\include\sqlfront.h" "\$(INTDIR)\time.obj" : \$(SOURCE) \$(DEP_CPP_TIME_) "\$(INTDIR)" \$(CPP) \$(CPP_PROJ) \$(SOURCE) # End Source File ##### # Begin Source File SOURCE=. \src\tpccldr.c DEP_CPP_TPCCL=\ ".\src\tpcc.h"\ "mssql\dblib\include\sqlldb.h"\ "mssql\dblib\include\sqlfront.h" "\$(INTDIR)\tpccldr.obj" : \$(SOURCE) \$(DEP_CPP_TPCCL) "\$(INTDIR)" \$(CPP) \$(CPP_PROJ) \$(SOURCE) # End Source File ##### # Begin Source File SOURCE=. \src\getargs.c DEP_CPP_GETAR=\ ".\src\tpcc.h"\ "mssql\dblib\include\sqlldb.h"\ "mssql\dblib\include\sqlfront.h" "\$(INTDIR)\getargs.obj" : \$(SOURCE) \$(DEP_CPP_GETAR) "\$(INTDIR)" \$(CPP) \$(CPP_PROJ) \$(SOURCE) # End Source File # End Target # End Project</pre>
--	---

#####

Appendix C: Tunable Parameters

Microsoft Windows 2000 Server Configuration Parameters

Microsoft Windows 2000 Server Services

Server Configuration Parameters

Microsoft Windows 2000 Advance Server Configuration
The following services were set as manual on the server:

- Alerter
- Computer Browser
- DHCP Client
- Distributed File System
- Distributed Link Tracking Client
- DNS Client
- IPSEC Policy Agent
- License Logging Service
- Messenger
- Print Spooler
- Remote Registry Service
- Removable Storage
- RUN as Service
- System Event Notification
- Task Scheduler

Microsoft SQL Server Startup Parameters

Microsoft SQL Server 2000 Startup Parameters

c:\Program Files\Microsoft SQL Server\MSSQL\Binn\sqlservr.exe -c -x -t3502 -g90

Where:

- c Start SQL Server independent of the Service Control Manager
- x Disable the keeping of CPU time and cache hit ratio statistics
- t3502 Writes a message to the SQL Server Errorlog showing the beginning and ending time of each checkpoint
- g90 Specifies the amount of memory that is set aside for allocations not from the buffer pool

Microsoft SQL Server 2000 Configuration Parameters

1> 2> 3> 4> 5> 6> 7> 8> 9> 10> 11>
-- File: VERSION.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.22
-- Copyright Microsoft, 2001
-- Purpose: Returns SQL Server version string

```
print " "  
select convert(char(30), getdate(),9)  
print " "
```

Jul 11 2002 10:22:21:903AM

(1 row affected)

```
1> 2> 3>  
select @@version
```

Microsoft SQL Server 2000 - 8.00.533 (Intel X86)
Nov 11 2001 22:04:31
Cop
yright (c) 1988-2000 Microsoft Corporation
Enterprise Edition on Windo
ws NT 5.0 (Build 2195: Service Pack 2)

(1 row affected)

```
1> 2>  
1> 2> 3> 4> 5> 6> 7> 8> 9> 10>  
-- File: CONFIG.SQL  
-- Microsoft TPC-C Benchmark Kit Ver. 4.22  
-- Copyright Microsoft, 2001  
-- Purpose: Collects SQL Server configuration parameters
```

```
print " "  
select convert(char(30), getdate(),9)  
print " "
```

Jul 11 2002 10:22:22:733AM

(1 row affected)

1> 2> 3> DBCC execution completed. If DBCC printed error messages, contact your system administrator.
Configuration option 'show advanced options' changed from 1 to 1. Run the RECONFIGURE statement to install.

```
sp_configure "show advanced",1  
1> 2> reconfigure with override  
1> 2> sp_configure
```

name	minimum	maximum
config_value run_value		
affinity mask	-2147483648	2147483647
255 255		
allow updates	0	1 0
0		
awe enabled	0	1 1
1		
c2 audit mode	0	1 0
0		
cost threshold for parallelism	0	32767
5 5		
cursor threshold	-1	2147483647
-1 -1		
default full-text language	0	2147483647
1033 1033		
default language	0	9999 0
0		

fill factor (%)	0	100	0
0			
index create memory (KB)	704	2147483647	
0 0			
lightweight pooling	0	1	1
1			
locks	5000	2147483647	0
0			
max degree of parallelism	0	32	1
1			
max server memory (MB)	4	2147483647	
2147483647 2147483647			
max text repl size (B)	0	2147483647	
65536 65536			
max worker threads	32	32767	
425 425			
media retention	0	365	0
0			
min memory per query (KB)	512	2147483647	
1024 1024			
min server memory (MB)	0	2147483647	
0 0			
nested triggers	0	1	1
1			
network packet size (B)	512	65536	
4096 4096			
open objects	0	2147483647	0
0			
priority boost	0	1	1
1			
query governor cost limit	0	2147483647	
0 0			
query wait (s)	-1	2147483647	-1
-1			
recovery interval (min)	0	32767	
54 54			
remote access	0	1	1
1			
remote login timeout (s)	0	2147483647	
20 20			
remote proc trans	0	1	0
0			
remote query timeout (s)	0	2147483647	
600 600			
scan for startup procs	0	1	0
0			
set working set size	0	1	0
0			
show advanced options	0	1	1
1			
two digit year cutoff	1753	9999	
2049 2049			
user connections	0	32767	0
0			
user options	0	32767	0
0			

1>

Microsoft Windows 2000 Server System Information Report

System Information report written at: 07/11/2002 02:24:12 PM
[System Information]

[Following are sub-categories of this main category]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Advanced Server
Version	5.0.2195 Service Pack 2 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	TPCC
System Manufacturer	IBM
System Model	eServer xSeries 360 -[86861RX]-
System Type	X86-based PC
Processor	x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1600 Mhz
Processor	x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1600 Mhz
Processor	x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1600 Mhz
Processor	x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1600 Mhz
Processor	x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1600 Mhz
Processor	x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1600 Mhz
Processor	x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1600 Mhz
BIOS Version	IBM BIOS Ver 1.04
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition1
Locale	United States
User Name	TPCC\Administrator
Time Zone	Eastern Daylight Time
Total Physical Memory	7,061,636 KB
Available Physical Memory	6,831,540 KB
Total Virtual Memory	16,083,852 KB
Available Virtual Memory	15,698,680 KB
Page File Space	9,022,216 KB
Page File	C:\pagefile.sys

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

Resource	Device
No conflicted/shared resources	

[DMA]

Channel	Device	Status
2	Standard floppy disk controller	OK
4	Direct memory access controller	OK

[Forced Hardware]

Device	PNP Device ID
No Forced Hardware	

[I/O]

Address Range	Device	Status
0x0000-0x1FFF	PCI bus	OK
0x0000-0x1FFF	Direct memory access controller	OK
0x03B0-0x03BB	S3 Inc. Savage4	OK
0x03C0-0x03DF	S3 Inc. Savage4	OK
0x1800-0x187F	Other PCI Bridge Device	OK
0x1880-0x18BF	Intel 8255x-based PCI Ethernet Adapter (10/100)	OK
0x1900-0x19FF	Adaptec AIC-7892 - Ultra160 SCSI	OK
0x0A79-0x0A79	ISAPNP Read Data Port	OK
0x0279-0x0279	ISAPNP Read Data Port	OK
0x02F4-0x02F7	ISAPNP Read Data Port	OK
0x002E-0x002F	Motherboard resources	OK
0x0438-0x0439	Motherboard resources	OK

0x0430-0x0437	Motherboard resources	OK
0x0060-0x0060	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard	OK
0x0064-0x0064	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard	OK
0x03F0-0x03F5	Standard floppy disk controller	OK
0x03F7-0x03F7	Standard floppy disk controller	OK
0x03F8-0x03FF	Motherboard resources	OK
0x0020-0x0021	Advanced programmable interrupt controller	OK
0x00A0-0x00A1	Advanced programmable interrupt controller	OK
0x0080-0x008F	Direct memory access controller	OK
0x00C0-0x00DF	Direct memory access controller	OK
0x0040-0x0043	System timer	OK
0x0070-0x0073	System CMOS/real time clock	OK
0x0061-0x0061	System speaker	OK
0x0374-0x0375	Motherboard resources	OK
0x0377-0x0377	Motherboard resources	OK
0x0490-0x04AF	Motherboard resources	OK
0x040B-0x040B	Motherboard resources	OK
0x04D0-0x04D1	Motherboard resources	OK
0x04D6-0x04D6	Motherboard resources	OK
0x0600-0x0600	Motherboard resources	OK
0x0C00-0x0C01	Motherboard resources	OK
0x0C14-0x0C14	Motherboard resources	OK
0x0C49-0x0C49	Motherboard resources	OK
0x0C4A-0x0C4A	Motherboard resources	OK
0x0C52-0x0C52	Motherboard resources	OK
0x0CD6-0x0CD7	Motherboard resources	OK
0x0F50-0x0F58	Motherboard resources	OK
0x0700-0x070F	Standard Dual Channel PCI IDE Controller	OK
0x01F0-0x01F7	Primary IDE Channel	OK
0x03F6-0x03F6	Primary IDE Channel	OK
0x0170-0x0177	Secondary IDE Channel	OK
0x0376-0x0376	Secondary IDE Channel	OK
0x4000-0x7FFF	PCI bus	OK
0x4000-0x7FFF	DEC 21154 PCI to PCI bridge	OK
0x4000-0x7FFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0x5000-0x5FFF	DEC 21154 PCI to PCI bridge	OK
0x5000-0x5FFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0x2000-0x3FFF	PCI bus	OK
0x2000-0x3FFF	DEC 21154 PCI to PCI bridge	OK
0x2000-0x3FFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0x3000-0x3FFF	DEC 21154 PCI to PCI bridge	OK
0x3000-0x3FFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK

[IRQs]

IRQ Number	Device
36	Microsoft ACPI-Compliant System
39	S3 Inc. Savage4
4	Other PCI Bridge Device
42	Intel 8255x-based PCI Ethernet Adapter (10/100)
41	Adaptec AIC-7892 - Ultra160 SCSI
1	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
12	PS/2 Compatible Mouse
6	Standard floppy disk controller
8	System CMOS/real time clock
14	Primary IDE Channel
16	Standard OpenHCD USB Host Controller
43	Mylex eXtremeRAID 2000 Disk Array Controller
19	Mylex eXtremeRAID 2000 Disk Array Controller
29	Mylex eXtremeRAID 2000 Disk Array Controller
33	Mylex eXtremeRAID 2000 Disk Array Controller

[Memory]

Range	Device	Status
0xF0000000-0xF7FFFFFF	PCI bus	OK
0xF0000000-0xF7FFFFFF	S3 Inc. Savage4	OK
0xF8000000-0xFBFFFFFF	PCI bus	OK
0xFBFB80000-0xFBFFFFFF	S3 Inc. Savage4	OK
0xA0000-0xBFFFF	S3 Inc. Savage4	OK
0xFBC00000-0xFBFFFFFF	Other PCI Bridge Device	OK
0xFBFB7F000-0xFBFB7FFFF	Intel 8255x-based PCI Ethernet Adapter (10/100)	OK
0xFBE00000-0xFBEFFFFFF	Intel 8255x-based PCI Ethernet Adapter (10/100)	OK
0xFBFB7E000-0xFBFB7EFFF	Adaptec AIC-7892 - Ultra160 SCSI	OK
0xFBFB7D000-0xFBFB7DFFF	Standard OpenHCD USB Host Controller	OK
0xE4000000-0xE7FFFFFF	PCI bus	OK
0xE4000000-0xE7FFFFFF	DEC 21154 PCI to PCI bridge	OK
0xE4000000-0xE7FFFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xE8000000-0xEFFFFFFF	PCI bus	OK
0xE8000000-0xEFFFFFFF	DEC 21154 PCI to PCI bridge	OK
0xE8000000-0xEFFFFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xEC000000-0xEE3FFFFF	DEC 21154 PCI to PCI bridge	OK
0xEC000000-0xEE3FFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xE6000000-0xE7FFFFFF	DEC 21154 PCI to PCI bridge	OK
0xE6000000-0xE7FFFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xD8000000-0xDFFFFFFF	PCI bus	OK
0xD8000000-0xDFFFFFFF	DEC 21154 PCI to PCI bridge	OK
0xD8000000-0xDFFFFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xE0000000-0xE3FFFFFF	PCI bus	OK
0xE0000000-0xE3FFFFFF	DEC 21154 PCI to PCI bridge	OK
0xE0000000-0xE3FFFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xDC000000-0xDE3FFFFF	DEC 21154 PCI to PCI bridge	OK
0xDC000000-0xDE3FFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xDA000000-0xDBFFFFFF	DEC 21154 PCI to PCI bridge	OK
0xDA000000-0xDBFFFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK

[Components]

[Following are sub-categories of this main category]

[Multimedia]

[Following are sub-categories of this main category]

[Audio Codecs]

Codec	Manufacturer	Description	Status	File
Version	Size	Creation Date		
c:\winnt\system32\iac25_32.ax	Intel Corporation	Indeo® audio software		
OK	C:\WINNT\System32\IAC25_32.AX	2.05.53	195.00 KB	
(199,680 bytes)	12/7/1999 6:00:00 AM			
c:\winnt\system32\lhacm.acm	Microsoft Corporation			
OK	C:\WINNT\System32\LHACM.ACM	4.4.3385	33.27 KB	
(34,064 bytes)	6/20/2002 10:44:37 AM			
c:\winnt\system32\msg723.acm	Microsoft Corporation			
OK	C:\WINNT\System32\MSG723.ACM	4.4.3385	106.77 KB	
(109,328 bytes)	6/20/2002 10:44:36 AM			

```

c:\winnt\system32\tssoft32.acm  DSP GROUP, INC.          OK
C:\WINNT\System32\TSSOFT32.ACM  1.01          9.27 KB (9,488 bytes)
12/7/1999 6:00:00 AM
c:\winnt\system32\msadp32.acm  Microsoft Corporation
OK      C:\WINNT\System32\MSADP32.ACM  5.00.2134.1
14.77 KB (15,120 bytes)          12/7/1999 6:00:00 AM
c:\winnt\system32\msgsm32.acm  Microsoft Corporation
OK      C:\WINNT\System32\MSGSM32.ACM  5.00.2134.1
22.27 KB (22,800 bytes)          12/7/1999 6:00:00 AM
c:\winnt\system32\imaadp32.acm Microsoft Corporation
OK      C:\WINNT\System32\IMAADP32.ACM  5.00.2134.1
16.27 KB (16,656 bytes)          12/7/1999 6:00:00 AM
c:\winnt\system32\msg711.acm   Microsoft Corporation
OK      C:\WINNT\System32\MSG711.ACM   5.00.2134.1
10.27 KB (10,512 bytes)          12/7/1999 6:00:00 AM

```

[Video Codecs]

Codec Version	Manufacturer	Description	Status	File
c:\winnt\system32\ir50_32.dll	Intel Corporation	Indeo® video 5.10	OK	C:\WINNT\System32\IR50_32.DLL
737.50 KB (755,200 bytes)		12/7/1999 6:00:00 AM		
c:\winnt\system32\msh261.drv	Microsoft Corporation		OK	C:\WINNT\System32\MSH261.DRV
167,696 bytes)		6/20/2002 10:44:37 AM		
c:\winnt\system32\msh263.drv	Microsoft Corporation		OK	C:\WINNT\System32\MSH263.DRV
258,320 bytes)		6/20/2002 10:44:09 AM		
c:\winnt\system32\msrle32.dll	Microsoft Corporation		OK	C:\WINNT\System32\MSRLE32.DLL
10.77 KB (11,024 bytes)		12/7/1999 6:00:00 AM		
c:\winnt\system32\ir32_32.dll	Intel(R) Corporation		OK	C:\WINNT\System32\IR32_32.DLL
199,168 bytes)		12/7/1999 6:00:00 AM		
c:\winnt\system32\msvidc32.dll	Microsoft Corporation		OK	C:\WINNT\System32\MSVIDC32.DLL
27.27 KB (27,920 bytes)		12/7/1999 6:00:00 AM		
c:\winnt\system32\iccvid.dll	Radius Inc.		OK	C:\WINNT\System32\ICCVID.DLL
108.00 KB (110,592 bytes)		12/7/1999 6:00:00 AM		

[CD-ROM]

Item	Value
Drive	D:
Description	CD-ROM Drive
Media Loaded	False
Media Type	CD-ROM
Name	TEAC CD-224E
Manufacturer	(Standard CD-ROM drives)
Status	OK
Transfer Rate	Not Available
SCSI Target ID	0
PNP Device ID	IDE\CDROMTEAC_CD-224E_2.7B_5&326853DD&0&0.0.0

[Sound Device]

Item	Value
No sound devices	

[Display]

Item	Value
Name	S3 Inc. Savage4

PNP Device ID	
PCI\VEN_5333&DEV_8A22&SUBSYS_01C51014&REV_06\3&267A616A&0&08	
Adapter Type	S3 Savage4, S3 compatible
Adapter Description	S3 Inc. Savage4
Adapter RAM	8.00 MB (8,388,608 bytes)
Installed Drivers	s3sav4.dll
Driver Version	5.01.840.0001
INF File	s3sav4.inf (S3Inc section)
Color Planes	1
Color Table Entries	65536
Resolution	1024 x 768 x 60 hertz
Bits/Pixel	16

[Infrared]

Item	Value
No infrared devices	

[Input]

[Following are sub-categories of this main category]

[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&23FD4C84&0
NumberOfFunctionKeys	12

[Pointing Device]

Item	Value
Hardware Type	PS/2 Compatible Mouse
Number of Buttons	3
Status	OK
PNP Device ID	ACPI\PNP0F13\4&23FD4C84&0
Power Management Supported	False
Double Click Threshold	6
Handedness	Right Handed Operation

[Modem]

Item	Value
No modems	

[Network]

[Following are sub-categories of this main category]

[Adapter]

Item	Value
Name	[00000000] Intel 8255x-based PCI Ethernet Adapter (10/100)
Adapter Type	Ethernet 802.3
Product Name	Intel 8255x-based PCI Ethernet Adapter (10/100)
Installed	True
PNP Device ID	PCI\VEN_8086&DEV_1229&SUBSYS_024D1014&REV_08\3&267A616A&0&18
Last Reset	7/11/2002 7:04:05 AM
Index	0

Service Name E100B
 IP Address 192.168.132.252
 IP Subnet 255.255.255.0
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:02:55:AC:15:9F
 Service Name E100B
 IRQ Number 42
 I/O Port 0x1880-0x18BF
 Driver c:\winnt\system32\drivers\e100bnt5.sys (119056, 5.40.11.0000)

Name [00000001] RAS Async Adapter
 Adapter Type Not Available
 Product Name RAS Async Adapter
 Installed True
 PNP Device ID Not Available
 Last Reset 7/11/2002 7:04:05 AM
 Index 1
 Service Name AsyncMac
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Service Name Not Available

Name [00000002] WAN Miniport (L2TP)
 Adapter Type Not Available
 Product Name WAN Miniport (L2TP)
 Installed True
 PNP Device ID ROOT\MS_L2TPMINIPORT\0000
 Last Reset 7/11/2002 7:04:05 AM
 Index 2
 Service Name Rasl2tp
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Service Name Rasl2tp
 Driver c:\winnt\system32\drivers\rasl2tp.sys (50800, 5.00.2179.1)

Name [00000003] WAN Miniport (PPTP)
 Adapter Type Wide Area Network (WAN)
 Product Name WAN Miniport (PPTP)
 Installed True
 PNP Device ID ROOT\MS_PPTPMINIPORT\0000
 Last Reset 7/11/2002 7:04:05 AM
 Index 3
 Service Name PptpMiniport
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 50:50:54:50:30:30
 Service Name PptpMiniport

Driver c:\winnt\system32\drivers\raspttp.sys (47856, 5.00.2160.1)

Name [00000004] Direct Parallel
 Adapter Type Not Available
 Product Name Direct Parallel
 Installed True
 PNP Device ID ROOT\MS_PTMINIPORT\0000
 Last Reset 7/11/2002 7:04:05 AM
 Index 4
 Service Name Raspti
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Service Name Raspti
 Driver c:\winnt\system32\drivers\raspti.sys (16880, 5.00.2146.1)

Name [00000005] WAN Miniport (IP)
 Adapter Type Not Available
 Product Name WAN Miniport (IP)
 Installed True
 PNP Device ID ROOT\MS_NDISWANIP\0000
 Last Reset 7/11/2002 7:04:05 AM
 Index 5
 Service Name NdisWan
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Service Name NdisWan
 Driver c:\winnt\system32\drivers\ndiswan.sys (90096, 5.00.2195.2779)

[Protocol]

Item	Value
Name	MSAFD Tcpip [TCP/IP]
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	16 bytes
MaximumMessageSize	0 bytes
MessageOriented	False
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	True
SupportsGracefulClosing	True
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Item	Value
Name	MSAFD Tcpip [UDP/IP]
ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	16 bytes
MaximumMessageSize	65467 bytes

MessageOriented True
 MinimumAddressSize 16 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting True

Name RSVP UDP Service Provider
 ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 16 bytes
 MaximumMessageSize 65467 bytes
 MessageOriented True
 MinimumAddressSize 16 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption True
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting True

Name RSVP TCP Service Provider
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 16 bytes
 MaximumMessageSize 0 bytes
 MessageOriented False
 MinimumAddressSize 16 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption True
 SupportsExpeditedData True
 SupportsGracefulClosing True
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{7E9DF517-8EC6-4125-9DB2-7978E0BEEDB6}]
 SEQUENCE 0
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize 20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{7E9DF517-8EC6-4125-9DB2-7978E0BEEDB6}]
 DATAGRAM 0
 ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize 20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{96C73D49-CA34-4DBB-9BAD-4F465F14AED7}]
 SEQUENCE 1
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize 20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{96C73D49-CA34-4DBB-9BAD-4F465F14AED7}]
 DATAGRAM 1
 ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize 20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{17FEBBFC-32E4-4CEF-AB31-D9183FC86644}]
 SEQUENCE 2
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes

MessageOriented True
 MinimumAddressSize 20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{17FEBBFC-32E4-4CEF-AB31-D9183FC86644}]
 DATAGRAM 2

ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize 20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

[WinSock]

Item	Value
File	c:\winnt\system32\winsock.dll
Version	3.10
Size	2.80 KB (2,864 bytes)

File	c:\winnt\system32\wsock32.dll
Version	5.00.2195.2871
Size	21.27 KB (21,776 bytes)

[Ports]

[Following are sub-categories of this main category]

[Serial]

Item	Value
No serial port information	

[Parallel]

Item	Value
No parallel port information	

[Storage]

[Following are sub-categories of this main category]

[Drives]

Item	Value
Drive A:	
Description	3 1/2 Inch Floppy Drive

Drive C:	
Description	Local Fixed Disk
Compressed	False
File System	NTFS
Size	8.46 GB (9,088,901,120 bytes)
Free Space	3.39 GB (3,643,248,640 bytes)
Volume Name	
Volume Serial Number	607FC777
Partition Disk #0, Partition #0	
Partition Size	8.46 GB (9,088,902,144 bytes)
Starting Offset	32256 bytes
Drive Description	Disk drive
Drive Manufacturer	(Standard disk drives)
Drive Model	IBM-PSG ST39103LC !# SCSI Disk Device
Drive BytesPerSector	512
Drive MediaLoaded	True
Drive MediaType	Fixed hard disk media
Drive Partitions	1
Drive SCSI Bus	0
Drive SCSI LogicalUnit	0
Drive SCSI Port	2
Drive SCSTargetId	12
Drive SectorsPerTrack	63
Drive Size	9097159680 bytes
Drive TotalCylinders	1106
Drive TotalSectors	17767890
Drive TotalTracks	282030
Drive TracksPerCylinder	255

Drive E:	
Description	Local Fixed Disk
Compressed	Not Available
File System	Not Available
Size	Not Available
Free Space	Not Available
Volume Name	Not Available
Volume Serial Number	Not Available

Drive F:	
Description	Local Fixed Disk
Compressed	Not Available
File System	Not Available
Size	Not Available
Free Space	Not Available
Volume Name	Not Available
Volume Serial Number	Not Available

Drive G:	
Description	Local Fixed Disk
Compressed	Not Available
File System	Not Available
Size	Not Available
Free Space	Not Available
Volume Name	Not Available
Volume Serial Number	Not Available

Drive H:	
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 I:	
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 Y:
 Description Local Fixed Disk
 Compressed False
 File System NTFS
 Size 146.48 GB (157,283,770,368 bytes)
 Free Space 12.54 GB (13,460,664,320 bytes)
 Volume Name Back_cool1
 Volume Serial Number 7C8DC2C1
 Partition Disk #1, Partition #0
 Partition Size 711.04 GB (763,470,489,600 bytes)
 Starting Offset 8225280 bytes
 Drive Description \\.\PHYSICALDRIVE1
 Drive Manufacturer Not Available
 Drive Model Not Available
 Drive BytesPerSector 512
 Drive MediaLoaded True
 Drive MediaType Fixed hard disk media
 Drive Partitions 2
 Drive SCSI Bus 4
 Drive SCSI Logical Unit 0
 Drive SCSI Port 3
 Drive SCSI Target ID 0
 Drive SectorsPerTrack 63
 Drive Size 763478714880 bytes
 Drive TotalCylinders 92821
 Drive TotalSectors 1491169365
 Drive TotalTracks 23669355
 Drive TracksPerCylinder 255

Drive Z:
 Description Local Fixed Disk
 Compressed False
 File System NTFS
 Size 146.48 GB (157,283,770,368 bytes)
 Free Space 12.50 GB (13,424,685,056 bytes)
 Volume Name Back_cool2
 Volume Serial Number D0A06548
 Partition Disk #2, Partition #0
 Partition Size 711.04 GB (763,470,489,600 bytes)
 Starting Offset 8225280 bytes
 Drive Description \\.\PHYSICALDRIVE2
 Drive Manufacturer Not Available
 Drive Model Not Available
 Drive BytesPerSector 512
 Drive MediaLoaded True
 Drive MediaType Fixed hard disk media
 Drive Partitions 2
 Drive SCSI Bus 4
 Drive SCSI Logical Unit 0
 Drive SCSI Port 4
 Drive SCSI Target ID 0
 Drive SectorsPerTrack 63
 Drive Size 763478714880 bytes
 Drive TotalCylinders 92821
 Drive TotalSectors 1491169365
 Drive TotalTracks 23669355
 Drive TracksPerCylinder 255

Drive J:
 Description Network Connection
 Provider Name \\192.168.132.253\c\$

[SCSI]

Item	Value
Name	Adaptec AIC-7892 - Ultra160 SCSI
Caption	Adaptec AIC-7892 - Ultra160 SCSI
Driver	adpu160m
Status	OK
PNP Device ID	PCI\VEN_9005&DEV_008F&SUBSYS_02011014&REV_02\3&267A616A&0&20
Device ID	PCI\VEN_9005&DEV_008F&SUBSYS_02011014&REV_02\3&267A616A&0&20
Device Map	Not Available
Index	Not Available
Max Number Controlled	Not Available
IRQ Number	41
I/O Port	0x1900-0x19FF
Driver	c:\winnt\system32\drivers\adpu160m.sys (88000, d4.10 (4.10.4000))

Name	Mylex eXtremeRAID 2000 Disk Array Controller
Caption	Mylex eXtremeRAID 2000 Disk Array Controller
Driver	dac2w2k
Status	OK
PNP Device ID	PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&1138865F&0&4008
Device ID	PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&1138865F&0&4008
Device Map	Not Available
Index	Not Available
Max Number Controlled	Not Available
IRQ Number	43
I/O Port	0x5000-0x5FFF
Driver	c:\winnt\system32\drivers\dac2w2k.sys (185584, 9.00-04)

Name	Mylex eXtremeRAID 2000 Disk Array Controller
Caption	Mylex eXtremeRAID 2000 Disk Array Controller
Driver	dac2w2k
Status	OK
PNP Device ID	PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&1A9D83F&0&4018
Device ID	PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&1A9D83F&0&4018
Device Map	Not Available
Index	Not Available
Max Number Controlled	Not Available
IRQ Number	19
I/O Port	0x4000-0x7FFF
Driver	c:\winnt\system32\drivers\dac2w2k.sys (185584, 9.00-04)

Name	Mylex eXtremeRAID 2000 Disk Array Controller
Caption	Mylex eXtremeRAID 2000 Disk Array Controller
Driver	dac2w2k
Status	OK
PNP Device ID	PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&235BDD1F&0&4008
Device ID	PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&235BDD1F&0&4008
Device Map	Not Available
Index	Not Available

Max Number Controlled Not Available
 IRQ Number 29
 I/O Port 0x3000-0x3FFF
 Driver c:\winnt\system32\drivers\dac2w2k.sys (185584, 9.00-04)

Name Mylex eXtremeRAID 2000 Disk Array Controller
 Caption Mylex eXtremeRAID 2000 Disk Array Controller
 Driver dac2w2k
 Status OK
 PNP Device ID
 PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&6CA15C9&0&4010
 Device ID
 PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&6CA15C9&0&4010
 Device Map Not Available
 Index Not Available
 Max Number Controlled Not Available
 IRQ Number 33
 I/O Port 0x2000-0x3FFF
 Driver c:\winnt\system32\drivers\dac2w2k.sys (185584, 9.00-04)

[Printing]

Name Port Name Server Name
 No printing information

[Problem Devices]

Device PNP Device ID Error Code
 Other PCI Bridge Device
 PCI\VEN_1014&DEV_010F&SUBSYS_01131014&REV_00\3&267A616A&0&10 28

[USB]

Device PNP Device ID
 Standard OpenHCD USB Host Controller
 PCI\VEN_1166&DEV_0220&SUBSYS_02201166&REV_04\3&267A616A&0&7A
 USB Root Hub USB\ROOT_HUB\4&372644EA&0

[Software Environment]

[Following are sub-categories of this main category]

[Drivers]

Name	Description	File	Type	Started	Start Mode
State	Status	Error Control	Accept Pause	Accept Stop	
abiosdsk	Abiosdsk	Not Available	Kernel Driver	False	
Disabled	Stopped	OK	Ignore	False	False
abp480n5	abp480n5	Not Available	Kernel Driver	False	
Disabled	Stopped	OK	Normal	False	False
acpi	Microsoft ACPI Driver	c:\winnt\system32\drivers\acpi.sys			
Kernel Driver	True	Boot	Running	OK	Normal
False	True				
acpiec	ACPIEC	c:\winnt\system32\drivers\acpiec.sys			
Driver	False	Disabled	Stopped	OK	Normal
False					False
adpu160m	adpu160m	c:\winnt\system32\drivers\adpu160m.sys			
Driver	True	Boot	Running	OK	Normal
True					False
afd	AFD Networking Support Environment				
c:\winnt\system32\drivers\afd.sys	Kernel Driver		True	Auto	
Running	OK	Normal	False	True	

aha154x	Aha154x	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
aic116x	aic116x	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
aic78u2	aic78u2	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
aic78xx	aic78xx	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
ami0nt	ami0nt	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
amsint	amsint	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
asc	asc	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
asc3350p	asc3350p	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
asc3550	asc3550	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
asynmac	RAS Asynchronous Media Driver			
c:\winnt\system32\drivers\asynmac.sys	Kernel Driver	False		
Manual	Stopped	OK	Normal	False
atapi	Standard IDE/ESDI Hard Disk Controller			
c:\winnt\system32\drivers\atapi.sys	Kernel Driver	True		
Boot	Running	OK	Normal	False
atdisk	Atdisk	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Ignore	False
atmarpc	ATM ARP Client Protocol			
c:\winnt\system32\drivers\atmarpc.sys	Kernel Driver	False		
Manual	Stopped	OK	Normal	False
audstub	Audio Stub Driver	c:\winnt\system32\drivers\audstub.sys		
Kernel Driver	True	Manual	Running	OK
False	True			Normal
beep	Beep	c:\winnt\system32\drivers\beep.sys		
Driver	True	System	Running	OK
True				Normal
buslogic	BusLogic	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
cd20xrnt	cd20xrnt	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
cdaudio	Cdaudio	c:\winnt\system32\drivers\cdaudio.sys		
Driver	False	System	Stopped	OK
False				Ignore
cdfs	Cdfs	c:\winnt\system32\drivers\cdfs.sys		
Driver	True	Disabled	Running	OK
True				Normal
cdrom	CD-ROM Driver	c:\winnt\system32\drivers\cdrom.sys		
Kernel Driver	True	System	Running	OK
False	True			Normal
changer	Changer	Not Available	Kernel Driver	False
System	Stopped	OK	Ignore	False
cpqarray	Cpqarray	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
cpqarry2	cpqarry2	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
cpqfcalm	cpqfcalm	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
cpqfws2e	cpqfws2e	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
dac2w2k	dac2w2k	c:\winnt\system32\drivers\dac2w2k.sys		
Driver	True	Boot	Running	OK
True				Normal
dac960nt	dac960nt	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
deckzpsx	deckzpsx	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
dfsdriver	DfsDriver	c:\winnt\system32\drivers\dfs.sys		
True	Boot	Running	OK	Normal
File System Driver	False	True		

disk	Disk Driver	c:\winnt\system32\drivers\disk.sys					
Kernel Driver	True	Boot	Running	OK	Normal		
False	True						
diskperf	Diskperf	c:\winnt\system32\drivers\diskperf.sys					
Driver	False	Disabled	Stopped	OK	Normal	False	
False							
dmboot	dmboot	c:\winnt\system32\drivers\dmboot.sys					
Driver	False	Disabled	Stopped	OK	Normal	False	
False							
dmio	Logical Disk Manager Driver						
	c:\winnt\system32\drivers\dmio.sys						
Boot	Running	OK	Normal	False	True	True	
dmload	dmload	c:\winnt\system32\drivers\dmload.sys					
Driver	True	Boot	Running	OK	Normal	False	
True							
e100b	Intel(R) PRO Adapter Driver						
	c:\winnt\system32\drivers\e100bnt5.sys						
Manual	Running	OK	Normal	False	True	True	
efs	EFS	c:\winnt\system32\drivers\efs.sys File System Driver					
True	Disabled	Running	OK	Normal	False	True	
fastfat	Fastfat	c:\winnt\system32\drivers\fastfat.sys					
Driver	True	Disabled	Running	OK	Normal	False	
True							
fd16_700	Fd16_700	Not Available					
Disabled	Stopped	OK	Normal	False	False	False	
fdc	Floppy Disk Controller Driver						
	c:\winnt\system32\drivers\fdc.sys						
Kernel Driver	True	True	Manual	Running	OK	Normal	
False	True						
fips	Fips	c:\winnt\system32\drivers\fips.sys					
Driver	True	Auto	Running	OK	Normal	False	
True							
fireport	fireport	Not Available					
Disabled	Stopped	OK	Normal	False	False	False	
flashpnt	flashpnt	Not Available					
Disabled	Stopped	OK	Normal	False	False	False	
flpydisk	Floppy Disk Driver	c:\winnt\system32\drivers\flpydisk.sys					
Kernel Driver	True	True	Manual	Running	OK	Normal	
False	True						
ftdisk	Volume Manager Driver						
	c:\winnt\system32\drivers\ftdisk.sys						
Boot	Running	OK	Normal	False	True	True	
gamdrv	gamdrv	c:\winnt\system32\drivers\gamdrv.sys					
Driver	True	Boot	Running	OK	Normal	False	
True							
gpc	Generic Packet Classifier						
	c:\winnt\system32\drivers\msgpc.sys						
Manual	Running	OK	Normal	False	True	True	
i8042prt	i8042 Keyboard and PS/2 Mouse Port Driver	c:\winnt\system32\drivers\i8042prt.sys					
System	Running	OK	Normal	False	True	True	
ibmhp	IBMHPA	c:\winnt\system32\drivers\ibmhp.sys					
Driver	True	Manual	Running	OK	Normal	False	
True							
ibmhp	IBM Active PCI Filter Driver						
	c:\winnt\system32\drivers\ibmhp.sys						
Boot	Running	OK	Normal	False	True	True	
ini910u	ini910u	Not Available					
Disabled	Stopped	OK	Normal	False	False	False	
intelide	IntelIde	Not Available					
Disabled	Stopped	OK	Normal	False	False	False	
ipfilterdriver	IP Traffic Filter Driver						
	c:\winnt\system32\drivers\ipfltdrv.sys						
Manual	Stopped	OK	Normal	False	False	False	
ipinip	IP in IP Tunnel Driver: c:\winnt\system32\drivers\ipinip.sys						
Kernel Driver	False	Manual	Stopped	OK	Normal	Normal	
False	False						
ipnat	IP Network Address Translator						
	c:\winnt\system32\drivers\ipnat.sys						
Kernel Driver	False	Manual	Stopped	OK	Normal	Normal	
False	False						
ipsec	IPSEC driver						
	c:\winnt\system32\drivers\ipsec.sys						
Kernel Driver	False	Manual	Stopped	OK	Normal	Normal	
False	False						
ipsraidn	ipsraidn	Not Available					
Disabled	Stopped	OK	Normal	False	False	False	
isapnp	PnP ISA/EISA Bus Driver						
	c:\winnt\system32\drivers\isapnp.sys						
Boot	Running	OK	Critical	False	True	True	
kbdclass	Keyboard Class Driver						
	c:\winnt\system32\drivers\kbdclass.sys						
System	Running	OK	Normal	False	True	True	
ksecdd	KSecDD	c:\winnt\system32\drivers\ksecdd.sys					
Driver	True	Boot	Running	OK	Normal	False	
True							
lbrtdc	lbrtdc	Not Available					
System	Stopped	OK	Ignore	False	False	False	
lp6nds35	lp6nds35	Not Available					
Disabled	Stopped	OK	Normal	False	False	False	
macdisk	macdisk	c:\winnt\system32\drivers\mac2w2k.sys					
Driver	True	Boot	Running	OK	Normal	Kernel	
True						False	
mnmd	mnmd	c:\winnt\system32\drivers\mnmd.sys					
Driver	True	System	Running	OK	Ignore	False	
True							
modem	Modem	c:\winnt\system32\drivers\modem.sys					
Driver	False	Manual	Stopped	OK	Ignore	False	
False							
mouclass	Mouse Class Driver						
	c:\winnt\system32\drivers\mouclass.sys						
Kernel Driver	True	System	Running	OK	Normal	Normal	
False	True						
mountmgr	MountMgr	c:\winnt\system32\drivers\mountmgr.sys					
Driver	True	Boot	Running	OK	Normal	False	
True							
mrraid35x	mrraid35x	Not Available					
Disabled	Stopped	OK	Normal	False	False	False	
mrxsm	MRXSMB	c:\winnt\system32\drivers\mrxsm.sys					
Driver	True	System	Running	OK	Normal	False	
True							
msfs	Msfs	c:\winnt\system32\drivers\msfs.sys					
Driver	True	System	Running	OK	Normal	False	
True							
mskssrv	Microsoft Streaming Service Proxy						
	c:\winnt\system32\drivers\mskssrv.sys						
Manual	Stopped	OK	Normal	False	False	False	
mspclock	Microsoft Streaming Clock Proxy						
	c:\winnt\system32\drivers\mspclock.sys						
Manual	Stopped	OK	Normal	False	False	False	
mspqm	Microsoft Streaming Quality Manager Proxy						
	c:\winnt\system32\drivers\mspqm.sys						
Manual	Stopped	OK	Normal	False	False	False	
mup	Mup	c:\winnt\system32\drivers\mup.sys					
Driver	True	Boot	Running	OK	Normal	File System	
True						False	
ncrc710	Nrc710	Not Available					
Disabled	Stopped	OK	Normal	False	False	False	
ndis	NDIS System Driver						
	c:\winnt\system32\drivers\ndis.sys						
Kernel Driver	True	Boot	Running	OK	Normal	Normal	
False	True						
ndistapi	Remote Access NDIS TAPI Driver						
	c:\winnt\system32\drivers\ndistapi.sys						
Manual	Running	OK	Normal	False	True	True	
ndiswan	Remote Access NDIS WAN Driver						
	c:\winnt\system32\drivers\ndiswan.sys						
Manual	Running	OK	Normal	False	True	True	

ndproxy	NDIS Proxy	c:\winnt\system32\drivers\ndproxy.sys		q11080	q11080	Not Available		Kernel Driver		False			
Kernel Driver	True	Manual	Running	OK	Normal	Disabled	Stopped	OK	Normal		False	False	
False	True			q110wnt	Q110wnt	Not Available		Kernel Driver		False			
netbios	NetBIOS Interface	c:\winnt\system32\drivers\netbios.sys		Disabled	Stopped	OK	Normal	False	False				
File System	Driver	True	System	Running	OK	Normal	Normal	False	False	False			
False	True			q11240	q11240	Not Available		Kernel Driver					
netbt	NetBios over Tcpip	c:\winnt\system32\drivers\netbt.sys		Disabled	Stopped	OK	Normal	False	False	False			
Kernel Driver	True	System	Running	OK	Normal	q12100	q12100	Not Available			Kernel Driver		
False	True			rasacd	Remote Access Auto Connection Driver	Disabled	Stopped	OK	Normal	False	False		
netdetect	NetDetect	c:\winnt\system32\drivers\netdect.sys		System	Running	OK	Normal	False	True	True			
Driver	False	Manual	Stopped	OK	Normal	False	rasl2tp	WAN Miniport (L2TP)			Kernel Driver		
False				npfs	Npfs	c:\winnt\system32\drivers\npfs.sys		File System		True			
npfs	Driver	True	System	Running	OK	Normal	False	c:\winnt\system32\drivers\rasl2tp.sys	Kernel Driver				
True				ntfs	Ntfs	c:\winnt\system32\drivers\ntfs.sys		File System		Normal			
Driver	True	Disabled	Running	OK	Normal	False	raspti	Direct Parallel			c:\winnt\system32\drivers\raspti.sys		
True				null	Null	c:\winnt\system32\drivers\null.sys		Kernel		Manual			
Driver	True	System	Running	OK	Normal	False	rca	Microsoft Streaming Network Raw Channel Access			Kernel Driver		
True				nwlkflt	IPX Traffic Filter Driver	c:\winnt\system32\drivers\nwlkflt.sys		Kernel Driver	False	File System			
nwlkflt	Driver	True	System	Running	OK	Normal	False	rdbs	Rdbss		c:\winnt\system32\drivers\rdbs.sys		
Manual	Stopped	OK	Normal	False	False	False	True	System	Running	OK	Normal		
nwlkfld	IPX Traffic Forwarder Driver	c:\winnt\system32\drivers\nwlkfld.sys		Kernel Driver	False	False	rdpwd	RDPWD	c:\winnt\system32\drivers\rdpwd.sys	Kernel			
Manual	Stopped	OK	Normal	False	False	False	Driver	False	Manual	Stopped	OK	Ignore	False
openhci	Microsoft USB Open Host Controller Driver	c:\winnt\system32\drivers\openhci.sys		Kernel Driver	True	True	redbook	Digital CD Audio Playback Filter Driver		Kernel Driver	False		
Manual	Running	OK	Normal	False	True	True	System	Stopped	OK	Normal	False	False	
parallel	Parallel	c:\winnt\system32\drivers\parallel.sys	Kernel	False	False	False	s3inc	S3Inc	c:\winnt\system32\drivers\s3sav4m.sys	Kernel			
Driver	False	Auto	Stopped	OK	Ignore	False	Driver	True	Manual	Running	OK	Ignore	False
False				parport	Parport	c:\winnt\system32\drivers\parport.sys		Kernel		Kernel			
parport	Driver	False	Auto	Stopped	OK	Ignore	False	serial	Serial		c:\winnt\system32\drivers\serial.sys	Kernel	
False				partmgr	PartMgr	c:\winnt\system32\drivers\partmgr.sys		Kernel		Kernel			
partmgr	Driver	True	Boot	Running	OK	Normal	False	sfloppy	Sfloppy		c:\winnt\system32\drivers\sfloppy.sys	Kernel	
True				parvdm	ParVdm	c:\winnt\system32\drivers\parvdm.sys		Kernel		False			
parvdm	Driver	False	Auto	Stopped	OK	Ignore	False	sglfb	sglfb		Not Available	Kernel Driver	
False				pci	PCI Bus Driver	c:\winnt\system32\drivers\pci.sys		Kernel		False			
pci	Driver	True	Boot	Running	OK	Critical	False	simbad	Simbad		Not Available	Kernel Driver	
True				pcidump	PCIDump	Not Available		Kernel Driver	False	False			
pcidump	System	Stopped	OK	Ignore	False	False	simbad	Simbad	Not Available		Kernel Driver		
pciide	PCIIde	c:\winnt\system32\drivers\pciide.sys	Kernel	False	False	False	sparrow	Sparrow	Not Available	Kernel Driver	False		
Driver	True	Boot	Running	OK	Normal	False	spud	Special Purpose Utility Driver	c:\winnt\system32\drivers\spud.sys	Kernel			
True				pcmcia	Pcmcia	c:\winnt\system32\drivers\pcmcia.sys		Kernel		Normal			
pcmcia	Driver	False	Disabled	Stopped	OK	Normal	False	srv	Srv		c:\winnt\system32\drivers\srv.sys	File System Driver	
False				pdcomp	PDCOMP	Not Available		Kernel Driver	False	True			
pdcomp	Manual	Stopped	OK	Ignore	False	False	swenum	Software Bus Driver	c:\winnt\system32\drivers\swenum.sys		Kernel Driver		
pdframe	PDFFRAME	Not Available		Kernel Driver	False	False	False	True	True	Manual	Running	OK	Normal
False	Manual	Stopped	OK	Ignore	False	False	symc810	symc810	Not Available	Kernel Driver	False		
pdrel	PDRELI	Not Available		Kernel Driver	False	False	symc8xx	symc8xx	Not Available	Kernel Driver	False		
Manual	Stopped	OK	Ignore	False	False	False	sym_hi	sym_hi	Not Available	Kernel Driver	False		
pdrframe	PDRFRAME	Not Available		Kernel Driver	False	False	tdasync	TDASync	c:\winnt\system32\drivers\tdasync.sys	Kernel Driver	False		
False	Manual	Stopped	OK	Ignore	False	False	tdipx	TDIPX	c:\winnt\system32\drivers\tdipx.sys	Kernel			
pptpminiport	WAN Miniport (PPTP)		c:\winnt\system32\drivers\raspptp.sys		Kernel Driver	True	False	False	False	Manual	Stopped	OK	Ignore
Manual	Running	OK	Normal	False	True	True	False	False	False	Manual	Stopped	OK	Ignore
ptilink	Direct Parallel Link Driver		c:\winnt\system32\drivers\ptilink.sys		Kernel Driver	True	False	False	False	Manual	Stopped	OK	Ignore
Manual	Running	OK	Normal	False	True	True	False	False	False	Manual	Stopped	OK	Ignore

tdnetb	TDNETB	c:\winnt\system32\drivers\tdnetb.sys	Kernel
Driver	False	Manual Stopped OK Ignore	False
False			
tdpipe	TDPIPE	c:\winnt\system32\drivers\tdpipe.sys	Kernel
Driver	False	Manual Stopped OK Ignore	False
False			
tdspdx	TDSPX	c:\winnt\system32\drivers\tdspdx.sys	Kernel
Driver	False	Manual Stopped OK Ignore	False
False			
tdtcp	TDTCP	c:\winnt\system32\drivers\tdtcp.sys	Kernel
Driver	False	Manual Stopped OK Ignore	False
False			
termdd	Terminal Device Driver		
	c:\winnt\system32\drivers\termdd.sys	Kernel Driver	False
Disabled	Stopped	OK Normal	False False
tga	tga	Not Available	Kernel Driver
System	Stopped	OK Ignore	False False
udfs	Udfs	c:\winnt\system32\drivers\udfs.sys	File System
Driver	False	Disabled Stopped OK Normal	False
False			
ultra66	ultra66	Not Available	Kernel Driver
Disabled	Stopped	OK Normal	False False
update	Microcode Update Driver		
	c:\winnt\system32\drivers\update.sys	Kernel Driver	True
Manual	Running	OK Normal	False True
usbhub	Microsoft USB Standard Hub Driver		
	c:\winnt\system32\drivers\usbhub.sys	Kernel Driver	True
Manual	Running	OK Normal	False True
vgasave	VgaSave	c:\winnt\system32\drivers\vga.sys	Kernel
Driver	True	System Running OK	Ignore
True			False
wanarp	Remote Access IP ARP Driver		
	c:\winnt\system32\drivers\wanarp.sys	Kernel Driver	True
Manual	Running	OK Normal	False True
wdica	WDICA	Not Available	Kernel Driver
Manual	Stopped	OK Ignore	False

[Environment Variables]

Variable	Value	User Name
ComSpec	%SystemRoot%\system32\cmd.exe	<SYSTEM>
NUMBER_OF_PROCESSORS	8	<SYSTEM>
OS	Windows_NT	<SYSTEM>
Os2LibPath	%SystemRoot%\system32\os2\dll;	<SYSTEM>
Path	%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem; C:\Program Files\Microsoft SQL Server\MSSQL\BINN;C:\Program Files\Microsoft SQL Server\80\Tools\BINN <SYSTEM>	
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH	<SYSTEM>
PROCESSOR_ARCHITECTURE	x86	<SYSTEM>
PROCESSOR_IDENTIFIER	x86 Family 15 Model 1 Stepping 1,	
	GenuineIntel	<SYSTEM>
PROCESSOR_LEVEL	15	<SYSTEM>
PROCESSOR_REVISION	0101	<SYSTEM>
TEMP	%SystemRoot%\TEMP	<SYSTEM>
TMP	%SystemRoot%\TEMP	<SYSTEM>
windir	%SystemRoot%	<SYSTEM>
TEMP	%USERPROFILE%\Local Settings\Temp	TPCC\Administrator
TMP	%USERPROFILE%\Local Settings\Temp	TPCC\Administrator

[Jobs]

[Following are sub-categories of this main category]

[Print]

Document	Size	Owner	Notify	Status	Time Submitted
Start Time	Until Time	Elapsed Time	Pages Printed	Job ID	
Priority	Parameters	Driver Name	Print Processor	Host Print	
Queue	Data Type	Name			
Unknown	Unknown	Unknown	Unknown	Unknown	Unknown
Unknown	Unknown	Unknown	Unknown	Unknown	Unknown
Unknown	Unknown	Unknown	Unknown		
[Network Connections]					
Local Name	Remote Name	Type	Status	User Name	
J:	\\192.168.132.253\e\$	Disk	OK	TPCC\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	Not Available	0	0	Not
Available	Not Available	Not Available	Unknown	Unknown	
Unknown					
system	Not Available	8	8	0	1413120
Not Available	Unknown	Unknown	Unknown		
smss.exe	c:\winnt\system32\smss.exe	188	11	204800	
1413120	7/11/2002 11:04:27 AM	5.00.2195.2901	44.27 KB		
(45,328 bytes)	12/7/1999 6:00:00 AM				
csrss.exe	Not Available	216	13	Not Available	
Not Available	7/11/2002 11:04:33 AM	Unknown	Unknown		
Unknown					
winlogon.exe	c:\winnt\system32\winlogon.exe	236	13		
204800	1413120	7/11/2002 11:04:35 AM	5.00.2195.2953		
173.77 KB (177,936 bytes)	12/7/1999 6:00:00 AM				
services.exe	c:\winnt\system32\services.exe	264	9		
204800	1413120	7/11/2002 11:04:36 AM	5.00.2195.2780		
86.77 KB (88,848 bytes)	12/7/1999 6:00:00 AM				
lsass.exe	c:\winnt\system32\lsass.exe	276	9	204800	
1413120	7/11/2002 11:04:36 AM	5.00.2195.2964	32.77 KB		
(33,552 bytes)	12/7/1999 6:00:00 AM				
svchost.exe	c:\winnt\system32\svchost.exe	424	8		
204800	1413120	7/11/2002 11:04:39 AM	5.00.2134.1		
7.77 KB (7,952 bytes)	12/7/1999 6:00:00 AM				
msdtc.exe	c:\winnt\system32\msdtc.exe	464	8	204800	
1413120	7/11/2002 11:04:40 AM	1999.9.3421.3	6.77 KB		
(6,928 bytes)	6/20/2002 6:39:24 AM				
svchost.exe	c:\winnt\system32\svchost.exe	716	8		
204800	1413120	7/11/2002 11:04:43 AM	5.00.2134.1		
7.77 KB (7,952 bytes)	12/7/1999 6:00:00 AM				
ibmhpasv.exe	c:\winnt\system32\ibmhpasv.exe	728	8		
204800	1413120	7/11/2002 11:04:43 AM	5.0.3.2	15.56 KB	
(15,935 bytes)	6/10/2002 7:22:05 PM				
tcpvcs.exe	c:\winnt\system32\tcpvcs.exe	784	8		
204800	1413120	7/11/2002 11:04:44 AM	5.00.2134.1		
24.77 KB (25,360 bytes)	12/7/1999 6:00:00 AM				
winmgmt.exe	c:\winnt\system32\wbem\winmgmt.exe	800	8		
204800	1413120	7/11/2002 11:04:44 AM	1.50.1085.0029		
192.08 KB (196,685 bytes)	6/20/2002 1:58:14 PM				
inetinfo.exe	c:\winnt\system32\inetrv\inetinfo.exe	880	8		
204800	1413120	7/11/2002 11:04:48 AM	5.00.0984	14.27 KB	
(14,608 bytes)	6/20/2002 1:59:00 PM				
explorer.exe	c:\winnt\explorer.exe	1028	8	204800	
1413120	7/11/2002 11:04:54 AM	5.00.3315.2846	237.27 KB		
(242,960 bytes)	6/20/2002 1:58:10 PM				
svchost.exe	c:\winnt\system32\svchost.exe	1268	8		
204800	1413120	7/11/2002 11:05:05 AM	5.00.2134.1		
7.77 KB (7,952 bytes)	12/7/1999 6:00:00 AM				
cmd.exe	c:\winnt\system32\cmd.exe	1340	8	204800	
1413120	7/11/2002 11:05:06 AM	5.00.2195.2104	230.77 KB		
(236,304 bytes)	12/7/1999 6:00:00 AM				

cmd.exe	c:\winnt\system32\cmd.exe	704	8	204800		cabinet.dll	5.00.2147.1	54.77 KB (56,080 bytes)	12/7/1999
1413120	7/11/2002 11:07:32 AM	5.00.2195.2104	230.77 KB			6:00:00 AM	Microsoft Corporation		
(236,304 bytes)	12/7/1999 6:00:00 AM					c:\winnt\system32\cabinet.dll			
gamscsm.exe	c:\winnt\system32\gamscsm.exe	1096	8			msinfo32.dll	5.00.2177.1	312.27 KB (319,760 bytes)	
204800	1413120	7/11/2002 11:12:59 AM	Not Available			6/20/2002 10:44:32 AM	Microsoft Corporation	c:\program	
119.81 KB (122,684 bytes)	6/28/2002 8:42:16 AM					files\common files\microsoft shared\msinfo\msinfo32.dll			
gamserv.exe	c:\winnt\system32\gamserv\gamserv.exe	1244				riched20.dll	5.30.23.1205	421.27 KB (431,376 bytes)	
13	204800	1413120	7/11/2002 11:12:59 AM	Not		6/20/2002 1:58:03 PM	Microsoft Corporation		
Available	128.70 KB (131,793 bytes)	6/28/2002 8:42:16 AM				c:\winnt\system32\riched20.dll			
gamevent.exe	c:\winnt\system32\gamserv\gamevent.exe	388				riched32.dll	5.00.2134.1	3.77 KB (3,856 bytes)	12/7/1999
13	204800	1413120	7/11/2002 11:12:59 AM	Not		6:00:00 AM	Microsoft Corporation		
Available	88.71 KB (90,834 bytes)	6/28/2002 8:42:15 AM				c:\winnt\system32\riched32.dll			
gamevlog.exe	c:\winnt\system32\gamserv\gamevlog.exe	1180				els.dll	5.00.2175.1	151.27 KB (154,896 bytes)	12/7/1999
13	204800	1413120	7/11/2002 11:12:59 AM	Not		6:00:00 AM	Microsoft Corporation		
Available	192.35 KB (196,962 bytes)	6/28/2002 8:42:15 AM				c:\winnt\system32\els.dll			
sqlservr.exe	c:\program files\microsoft sql					ntmsmgr.dll	1,0,0,1	427.77 KB (438,032 bytes)	12/7/1999
server\mssql\binn\sqlservr.exe	1444	13	204800	1413120		6:00:00 AM	Microsoft Corporation and HighGround Systems, Inc.		
7/11/2002 12:57:04 PM	2000.080.0533.00	7.10 MB (7,446,609				c:\winnt\system32\ntmsmgr.dll			
bytes)	6/20/2002 12:31:16 PM					mmfutil.dll	1.50.1085.0000	32.06 KB (32,829 bytes)	12/7/1999
isqlw.exe	c:\program files\microsoft sql server\80\tools\binn\isqlw.exe					6:00:00 AM	Microsoft Corporation		
1388	8	204800	1413120	7/11/2002 12:57:23 PM		c:\winnt\system32\mmfutil.dll			
2000.080.0382.00	344.56 KB (352,828 bytes)	6/20/2002 12:31:37 PM				logdrive.dll	1.50.1085.0000	200.06 KB (204,863 bytes)	
mdm.exe	c:\winnt\system32\mdm.exe	1348	8	204800		12/7/1999 6:00:00 AM	Microsoft Corporation		
1413120	7/11/2002 1:08:04 PM	6.00.8424	121.29 KB (124,200			c:\winnt\system32\logdrive.dll			
bytes)	6/20/2002 6:41:40 AM					dfrgres.dll	5.00.2150.1	27.50 KB (28,160 bytes)	12/7/1999
mmc.exe	c:\winnt\system32\mmc.exe	1332	8	204800		6:00:00 AM	Executive Software International, Inc.		
1413120	7/11/2002 2:22:13 PM	5.00.2195.2301	589.27 KB			c:\winnt\system32\dfrgres.dll			
(603,408 bytes)	6/20/2002 1:57:50 PM					dfrgsnap.dll	5.00.2195.2104	41.77 KB (42,768 bytes)	
rsrvp.exe	c:\winnt\system32\rsrvp.exe	1600	8	204800		6/20/2002 1:57:43 PM	Executive Software International, Inc.		
1413120	7/11/2002 2:23:44 PM	5.00.2167.1	172.77 KB			c:\winnt\system32\dfrgsnap.dll			
(176,912 bytes)	12/7/1999 6:00:00 AM					dmdskres.dll	2195.2104.297.3	119.50 KB (122,368 bytes)	
						6/20/2002 1:57:43 PM	Microsoft Corp., VERITAS Software		
						c:\winnt\system32\dmdskres.dll			
						dmutil.dll	2195.2104.297.3	42.27 KB (43,280 bytes)	6/20/2002
						1:57:43 PM	VERITAS Software Corp.		
						c:\winnt\system32\dmutil.dll			
						ntmsapi.dll	5.00.1948.1	51.77 KB (53,008 bytes)	6/20/2002
						1:58:00 PM	Microsoft Corporation		
						c:\winnt\system32\ntmsapi.dll			
						dmdskmgr.dll	2215.2215.297.3	160.27 KB (164,112 bytes)	
						6/20/2002 1:57:43 PM	Microsoft Corp., VERITAS Software		
						c:\winnt\system32\dmdskmgr.dll			
						mycomput.dll	5.00.2134.1	107.77 KB (110,352 bytes)	
						12/7/1999 6:00:00 AM	Microsoft Corporation		
						c:\winnt\system32\mycomput.dll			
						mmcmdmgr.dll	5.00.2178.1	815.27 KB (834,832 bytes)	
						12/7/1999 6:00:00 AM	Microsoft Corporation		
						c:\winnt\system32\mmcmdmgr.dll			
						mfc42u.dll	6.00.8665.0	972.05 KB (995,384 bytes)	12/7/1999
						6:00:00 AM	Microsoft Corporation		
						c:\winnt\system32\mfc42u.dll			
						mmc.exe	5.00.2195.2301	589.27 KB (603,408 bytes)	6/20/2002
						1:57:50 PM	Microsoft Corporation		
						c:\winnt\system32\mmc.exe			
						mdm.exe	6.00.8424	121.29 KB (124,200 bytes)	6/20/2002 6:41:40 AM
						Microsoft Corporation	c:\winnt\system32\mdm.exe		
						dbmslpcn.dll	2000.080.0533.00	28.56 KB (29,244 bytes)	
						6/20/2002 12:31:19 PM	Microsoft Corporation		
						c:\winnt\system32\dbmslpcn.dll			
						dbnetlib.dll	2000.080.0528.00	84.08 KB (86,097 bytes)	
						6/20/2002 2:12:10 PM	Microsoft Corporation		
						c:\winnt\system32\dbnetlib.dll			
						sqlsrv32.rll	2000.080.0528.00	88.00 KB (90,112 bytes)	
						6/20/2002 2:12:12 PM	Microsoft Corporation		
						c:\winnt\system32\sqlsrv32.rll			
						sqlsrv32.dll	2000.080.0528.00	460.08 KB (471,121 bytes)	
						6/20/2002 2:12:12 PM	Microsoft Corporation		
						c:\winnt\system32\sqlsrv32.dll			

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
traffic.dll	5.00.2139.1	30.77 KB (31,504 bytes)			
6:00:00 AM	Microsoft Corporation				12/7/1999
c:\winnt\system32\traffic.dll					
rsrvp.exe	5.00.2167.1	172.77 KB (176,912 bytes)			12/7/1999
6:00:00 AM	Microsoft Corporation				
c:\winnt\system32\rsrvp.exe					
wbemprox.dll	1.50.1085.0045	40.08 KB (41,040 bytes)			
6/20/2002 1:58:14 PM	Microsoft Corporation				
c:\winnt\system32\wbem\wbemprox.dll					
rassapi.dll	5.00.2188.1	14.27 KB (14,608 bytes)			12/7/1999
6:00:00 AM	Microsoft Corporation				
c:\winnt\system32\rassapi.dll					
adsnt.dll	5.00.2195.2778	195.27 KB (199,952 bytes)			6/20/2002
1:57:38 PM	Microsoft Corporation				
c:\winnt\system32\adsnt.dll					
dbghelp.dll	5.00.2195.2104	159.27 KB (163,088 bytes)			
5/4/2001 12:05:02 PM	Microsoft Corporation				
c:\winnt\system32\dbghelp.dll					
localsec.dll	5.00.2195.2130	230.27 KB (235,792 bytes)			
6/20/2002 1:57:49 PM	Microsoft Corporation				
c:\winnt\system32\localsec.dll					
devmgr.dll	5.00.2166.1	215.77 KB (220,944 bytes)			12/7/1999
6:00:00 AM	Microsoft Corporation				
c:\winnt\system32\devmgr.dll					
filegmt.dll	5.00.2195.2165	287.27 KB (294,160 bytes)			
6/20/2002 1:57:46 PM	Microsoft Corporation				
c:\winnt\system32\filegmt.dll					
pdh.dll	5.00.2195.2739	147.77 KB (151,312 bytes)			6/20/2002
1:58:02 PM	Microsoft Corporation				
c:\winnt\system32\pdh.dll					
smlogcfg.dll	5.00.2195.2485	273.27 KB (279,824 bytes)			
6/20/2002 1:58:06 PM	Microsoft Corporation				
c:\winnt\system32\smlogcfg.dll					

sqllex.dll	2000.080.0194.00	148.06 KB (151,616 bytes)	6/20/2002 12:31:51 PM	Microsoft Corporation	c:\program files\microsoft sql server\80\tools\bin\sqllex.dll
odbc32.dll	3.520.7926.0	100.27 KB (102,672 bytes)	6/20/2002 2:12:05 PM	Microsoft Corporation	c:\winnt\system32\odbc32.dll
isqlw.rll	2000.080.0382.00	240.00 KB (245,760 bytes)	6/20/2002 12:31:37 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\resources\1033\isqlw.rll
sqlqry.rll	2000.080.0194.00	180.00 KB (184,320 bytes)	6/20/2002 12:31:37 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\resources\1033\sqlqry.rll
pfutil80.rll	2000.080.0382.00	144.00 KB (147,456 bytes)	6/20/2002 12:31:52 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\resources\1033\pfutil80.rll
pfclnt80.rll	2000.080.0194.00	28.00 KB (28,672 bytes)	6/20/2002 12:31:50 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\resources\1033\pfclnt80.rll
semsfc.rll	2000.080.0194.00	24.00 KB (24,576 bytes)	6/20/2002 12:31:51 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\resources\1033\semsfc.rll
sqlgui.rll	2000.080.0194.00	56.00 KB (57,344 bytes)	6/20/2002 12:31:51 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\resources\1033\sqlgui.rll
sqlsvc.rll	2000.080.0194.00	24.00 KB (24,576 bytes)	6/20/2002 12:31:50 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\resources\1033\sqlsvc.rll
odbcint.dll	3.520.7926.0	88.00 KB (90,112 bytes)	6/20/2002 2:12:06 PM	Microsoft Corporation	c:\winnt\system32\odbcint.dll
pfclnt80.dll	2000.080.0533.00	404.56 KB (414,272 bytes)	6/20/2002 12:31:50 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\pfclnt80.dll
semsfc.dll	2000.080.0533.00	224.56 KB (229,952 bytes)	6/20/2002 12:31:50 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\semsfc.dll
pfutil80.dll	2000.080.0533.00	272.56 KB (279,104 bytes)	6/20/2002 12:31:51 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\pfutil80.dll
sqlqry.dll	2000.080.0382.00	392.56 KB (401,984 bytes)	6/20/2002 12:31:37 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\sqlqry.dll
odbc32.dll	2000.080.0528.00	28.57 KB (29,252 bytes)	6/20/2002 2:12:12 PM	Microsoft Corporation	c:\winnt\system32\odbc32.dll
sqlsvc.dll	2000.080.0382.00	92.56 KB (94,784 bytes)	6/20/2002 12:31:50 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\sqlsvc.dll
odbc32.dll	3.520.7926.0	216.27 KB (221,456 bytes)	6/20/2002 2:12:05 PM	Microsoft Corporation	c:\winnt\system32\odbc32.dll
w95scm.dll	2000.080.0194.00	48.56 KB (49,728 bytes)	6/20/2002 12:31:50 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\w95scm.dll
sqlgui.dll	2000.080.0533.00	444.56 KB (455,232 bytes)	6/20/2002 12:31:50 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\sqlgui.dll
sqlresld.dll	2000.080.0382.00	28.56 KB (29,248 bytes)	6/20/2002 12:31:50 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\sqlresld.dll
sqlunirl.dll	2000.080.0380.00	176.56 KB (180,800 bytes)	4/9/2001 9:46:18 AM	Microsoft Corporation	c:\winnt\system32\sqlunirl.dll
isqlw.exe	2000.080.0382.00	344.56 KB (352,828 bytes)	6/20/2002 12:31:37 PM	Microsoft Corporation	files\microsoft sql server\80\tools\bin\isqlw.exe
ssmslpcn.dll	2000.080.0533.00	28.56 KB (29,244 bytes)	6/20/2002 12:31:19 PM	Microsoft Corporation	files\microsoft sql server\mssql\bin\ssmslpcn.dll
ssnmpn70.dll	2000.080.0533.00	24.56 KB (25,148 bytes)	6/20/2002 12:31:19 PM	Microsoft Corporation	files\microsoft sql server\mssql\bin\ssnmpn70.dll
ssnetlib.dll	2000.080.0533.00	84.56 KB (86,588 bytes)	6/20/2002 12:31:18 PM	Microsoft Corporation	files\microsoft sql server\mssql\bin\ssnetlib.dll
sqllevn70.rll	2000.080.0533.00	28.00 KB (28,672 bytes)	6/20/2002 12:31:19 PM	Microsoft Corporation	files\microsoft sql server\mssql\bin\resources\1033\sqllevn70.rll
msvcirt.dll	6.10.8637.0	76.05 KB (77,878 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\msvcirt.dll
sqlsort.dll	2000.080.0533.00	576.56 KB (590,396 bytes)	6/20/2002 12:31:19 PM	Microsoft Corporation	files\microsoft sql server\mssql\bin\sqlsort.dll
ums.dll	2000.080.0382.00	48.07 KB (49,228 bytes)	6/20/2002 12:31:18 PM	Microsoft Corporation	files\microsoft sql server\mssql\bin\ums.dll
opends60.dll	2000.080.0194.00	24.06 KB (24,639 bytes)	6/20/2002 12:31:18 PM	Microsoft Corporation	files\microsoft sql server\mssql\bin\opends60.dll
sqlservr.exe	2000.080.0533.00	7.10 MB (7,446,609 bytes)	6/20/2002 12:31:16 PM	Microsoft Corporation	files\microsoft sql server\mssql\bin\sqlservr.exe
gamevlog.exe	Not Available	192.35 KB (196,962 bytes)	6/28/2002 8:42:15 AM	Not Available	c:\winnt\system32\gamevlog.exe
gamevent.exe	Not Available	88.71 KB (90,834 bytes)	6/28/2002 8:42:15 AM	Not Available	c:\winnt\system32\gamevlog.exe
gamserv.exe	Not Available	128.70 KB (131,793 bytes)	6/28/2002 8:42:16 AM	Not Available	c:\winnt\system32\gamevlog.exe
gamserv.exe	Not Available	119.81 KB (122,684 bytes)	6/28/2002 8:42:16 AM	Not Available	c:\winnt\system32\gamevlog.exe
cmd.exe	5.00.2195.2104	230.77 KB (236,304 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\cmd.exe
tapisrv.dll	5.00.2195.2955	169.27 KB (173,328 bytes)	6/20/2002 1:58:07 PM	Microsoft Corporation	c:\winnt\system32\tapisrv.dll
thumbvw.dll	5.00.2920.0000	183.27 KB (187,664 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\thumbvw.dll
usp10.dll	1.0325.2195.2104	308.27 KB (315,664 bytes)	6/20/2002 1:58:08 PM	Microsoft Corporation	c:\winnt\system32\usp10.dll
imgutil.dll	5.00.3315.2870	30.77 KB (31,504 bytes)	6/20/2002 1:57:47 PM	Microsoft Corporation	c:\winnt\system32\imgutil.dll
msls31.dll	3.10.337.0	145.27 KB (148,752 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\msls31.dll
shdoclc.dll	5.00.3315.2879	324.50 KB (332,288 bytes)	6/20/2002 1:58:05 PM	Microsoft Corporation	c:\winnt\system32\shdoclc.dll
mshtml.dll	5.00.3315.2870	2.24 MB (2,345,232 bytes)	6/20/2002 1:57:52 PM	Microsoft Corporation	c:\winnt\system32\mshtml.dll
faxshell.dll	5.00.2134.1	8.27 KB (8,464 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\faxshell.dll
msacm32.dll	5.00.2134.1	65.27 KB (66,832 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\svstem32\msacm32.dll

avifil32.dll 5.00.2134.1	76.27 KB (78,096 bytes)	12/7/1999	httpext.dll 0.9.3940.21	435.27 KB (445,712 bytes)	6/20/2002
6:00:00 AM	Microsoft Corporation		1:58:59 PM	Microsoft Corporation	
c:\winnt\system32\avifil32.dll			c:\winnt\system32\inetrv\httpext.dll		
msvfw32.dll 5.00.2134.1	113.77 KB (116,496 bytes)		rpcproxy.dll 5.00.2195.2780	16.27 KB (16,656 bytes)	
12/7/1999 6:00:00 AM	Microsoft Corporation		6/20/2002 1:58:52 PM	Microsoft Corporation	
c:\winnt\system32\msvfw32.dll			c:\winnt\system32\rpcproxy\rpcproxy.dll		
docprop2.dll 5.00.2178.1	297.77 KB (304,912 bytes)		fpexedll.dll 4.0.2.4324	20.06 KB (20,541 bytes)	6/20/2002
12/7/1999 6:00:00 AM	Microsoft Corporation		1:58:51 PM	Microsoft Corporation	c:\program
c:\winnt\system32\docprop2.dll			files\common files\microsoft shared\web server extensions\40\bin\fpexedll.dll		
dsuixt.dll 5.00.2195.2779	107.77 KB (110,352 bytes)	6/20/2002	md5filt.dll 5.00.0984	32.77 KB (33,552 bytes)	6/20/2002 1:59:00 PM
1:57:44 PM	Microsoft Corporation		Microsoft Corporation	c:\winnt\system32\inetrv\md5filt.dll	
c:\winnt\system32\dsuixt.dll			gzip.dll 5.00.0984	30.27 KB (30,992 bytes)	6/20/2002 1:58:59 PM
dsquery.dll 5.00.2195.2854	153.27 KB (156,944 bytes)	6/20/2002	Microsoft Corporation	c:\winnt\system32\inetrv\gzip.dll	
1:57:44 PM	Microsoft Corporation		compfilt.dll 5.00.0984	22.77 KB (23,312 bytes)	6/20/2002
c:\winnt\system32\dsquery.dll			1:58:59 PM	Microsoft Corporation	
imm32.dll 5.00.2195.2821	94.27 KB (96,528 bytes)	6/20/2002	c:\winnt\system32\inetrv\compfilt.dll		
1:57:47 PM	Microsoft Corporation		sspfilt.dll 5.00.0984	43.27 KB (44,304 bytes)	6/20/2002 1:59:00 PM
c:\winnt\system32\imm32.dll			Microsoft Corporation	c:\winnt\system32\inetrv\sspfilt.dll	
jscrip.dll 5.1.0.5907	476.06 KB (487,481 bytes)	6/20/2002 1:57:49 PM	iscomlog.dll 5.00.0984	24.77 KB (25,360 bytes)	6/20/2002
Microsoft Corporation	c:\winnt\system32\jscrip.dll		1:59:00 PM	Microsoft Corporation	
wininet.dll 5.00.3315.1000	456.77 KB (467,728 bytes)	6/20/2002	c:\winnt\system32\inetrv\iscomlog.dll		
1:58:09 PM	Microsoft Corporation		lonsint.dll 5.00.0984	11.77 KB (12,048 bytes)	6/20/2002 1:59:00 PM
c:\winnt\system32\wininet.dll			Microsoft Corporation	c:\winnt\system32\inetrv\lonsint.dll	
msdbg.dll 6.00.8424	67.50 KB (69,120 bytes)	6/20/2002 6:41:40 AM	inetsloc.dll 5.00.0984	20.27 KB (20,752 bytes)	6/20/2002 1:57:47 PM
Microsoft Corporation	c:\winnt\system32\msdbg.dll		Microsoft Corporation	c:\winnt\system32\inetrv\inetsloc.dll	
pdm.dll 6.00.8424	179.27 KB (183,574 bytes)	6/20/2002 6:41:40 AM	iisfecnv.dll 5.00.0984	7.27 KB (7,440 bytes)	6/20/2002 6:40:01 AM
Microsoft Corporation	c:\winnt\system32\pdm.dll		Microsoft Corporation	c:\winnt\system32\inetrv\iisfecnv.dll	
mlang.dll 5.00.3103.1000	510.77 KB (523,024 bytes)	6/20/2002	isatq.dll 5.00.0984	60.27 KB (61,712 bytes)	6/20/2002 1:59:00 PM
1:57:50 PM	Microsoft Corporation		Microsoft Corporation	c:\winnt\system32\inetrv\isatq.dll	
c:\winnt\system32\mlang.dll			infocomm.dll 5.00.0984	238.27 KB (243,984 bytes)	6/20/2002
urlmon.dll 5.00.3315.1000	441.27 KB (451,856 bytes)	6/20/2002	1:59:00 PM	Microsoft Corporation	
1:58:08 PM	Microsoft Corporation		c:\winnt\system32\inetrv\infocomm.dll		
c:\winnt\system32\urlmon.dll			w3svc.dll 5.00.0984	343.27 KB (351,504 bytes)	6/20/2002 1:59:01 PM
browseic.dll 5.00.3315.2846	34.50 KB (35,328 bytes)		Microsoft Corporation	c:\winnt\system32\inetrv\w3svc.dll	
6/20/2002 1:57:39 PM	Microsoft Corporation		security.dll 5.00.2154.1	5.77 KB (5,904 bytes)	12/7/1999 6:00:00 AM
c:\winnt\system32\browseic.dll			Microsoft Corporation	c:\winnt\system32\security.dll	
linkinfo.dll 5.00.2134.1	15.77 KB (16,144 bytes)	12/7/1999	svccxt.dll 5.00.0984	39.77 KB (40,720 bytes)	6/20/2002 1:59:00 PM
6:00:00 AM	Microsoft Corporation		Microsoft Corporation	c:\winnt\system32\inetrv\svccxt.dll	
c:\winnt\system32\linkinfo.dll			admexs.dll 5.00.0984	27.77 KB (28,432 bytes)	6/20/2002 1:58:58 PM
powrprof.dll 5.00.3103.1000	13.27 KB (13,584 bytes)		Microsoft Corporation	c:\winnt\system32\inetrv\admexs.dll	
6/20/2002 1:58:02 PM	Microsoft Corporation		wamreg.dll 5.00.0984	45.77 KB (46,864 bytes)	6/20/2002 1:59:01 PM
c:\winnt\system32\powrprof.dll			Microsoft Corporation	c:\winnt\system32\inetrv\wamreg.dll	
batmeter.dll 5.00.3103.1000	20.27 KB (20,752 bytes)		metadata.dll 5.00.0984	68.77 KB (70,416 bytes)	6/20/2002
6/20/2002 1:57:39 PM	Microsoft Corporation		1:59:00 PM	Microsoft Corporation	
c:\winnt\system32\batmeter.dll			c:\winnt\system32\inetrv\metadata.dll		
stobject.dll 5.00.2195.2780	79.27 KB (81,168 bytes)	6/20/2002	iismap.dll 5.00.0984	55.77 KB (57,104 bytes)	6/20/2002 1:57:47 PM
1:58:07 PM	Microsoft Corporation		Microsoft Corporation	c:\winnt\system32\iismap.dll	
c:\winnt\system32\stobject.dll			nsepm.dll 5.00.0984	43.27 KB (44,304 bytes)	6/20/2002 1:59:00 PM
webcheck.dll 5.00.3315.1000	251.77 KB (257,808 bytes)		Microsoft Corporation	c:\winnt\system32\inetrv\nsepm.dll	
6/20/2002 1:58:08 PM	Microsoft Corporation		admwprox.dll 5.00.0984	31.77 KB (32,528 bytes)	6/20/2002
c:\winnt\system32\webcheck.dll			6:40:02 AM	Microsoft Corporation	
ntshrui.dll 5.00.2134.1	46.77 KB (47,888 bytes)	12/7/1999	c:\winnt\system32\admwprox.dll		
6:00:00 AM	Microsoft Corporation		coadmin.dll 5.00.0984	39.27 KB (40,208 bytes)	6/20/2002
c:\winnt\system32\ntshrui.dll			1:58:59 PM	Microsoft Corporation	
mydocs.dll 5.00.2920.0000	55.77 KB (57,104 bytes)	12/7/1999	c:\winnt\system32\inetrv\coadmin.dll		
6:00:00 AM	Microsoft Corporation		iisadmin.dll 5.00.0984	15.27 KB (15,632 bytes)	6/20/2002
c:\winnt\system32\mydocs.dll			1:58:59 PM	Microsoft Corporation	
browseui.dll 5.00.3315.2846	788.77 KB (807,696 bytes)		c:\winnt\system32\inetrv\iisadmin.dll		
6/20/2002 1:57:39 PM	Microsoft Corporation		rpcref.dll 5.00.0984	4.27 KB (4,368 bytes)	6/20/2002 1:59:00 PM
c:\winnt\system32\browseui.dll			Microsoft Corporation	c:\winnt\system32\inetrv\rpcref.dll	
shdocvw.dll 5.00.3315.2879	1.05 MB (1,104,144 bytes)		iisrtl.dll 5.00.0984	119.77 KB (122,640 bytes)	6/20/2002 1:57:47 PM
6/20/2002 1:58:05 PM	Microsoft Corporation		Microsoft Corporation	c:\winnt\system32\iisrtl.dll	
c:\winnt\system32\shdocvw.dll			inetinfo.exe 5.00.0984	14.27 KB (14,608 bytes)	6/20/2002
explorer.exe 5.00.3315.2846	237.27 KB (242,960 bytes)		1:59:00 PM	Microsoft Corporation	
6/20/2002 1:58:10 PM	Microsoft Corporation		c:\winnt\system32\inetrv\inetinfo.exe		
c:\winnt\explorer.exe			wshnetbs.dll 5.00.2134.1	7.77 KB (7,952 bytes)	12/7/1999
iislog.dll 5.00.0984	75.27 KB (77,072 bytes)	6/20/2002 1:58:59 PM	6:00:00 AM	Microsoft Corporation	
Microsoft Corporation	c:\winnt\system32\inetrv\iislog.dll		c:\winnt\system32\wshnetbs.dll		

ntmarta.dll 5.00.2195.2862 98.77 KB (101,136 bytes) 6/20/2002 1:58:00 PM Microsoft Corporation	es.dll 2000.2.3471.1 222.27 KB (227,600 bytes) 6/20/2002 1:57:45 PM Microsoft Corporation
c:\winnt\system32\ntmarta.dll	c:\winnt\system32\es.dll
perfos.dll 5.00.2155.1 21.27 KB (21,776 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation	iashlpr.dll 5.00.2184.1 33.27 KB (34,064 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation
c:\winnt\system32\perfos.dll	c:\winnt\system32\iashlpr.dll
provthrd.dll 1.50.1085.0000 68.07 KB (69,708 bytes) 6/20/2002 10:44:23 AM Microsoft Corporation	iasacct.dll 5.00.2134.1 28.27 KB (28,944 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\provthrd.dll	c:\winnt\system32\iasacct.dll
ntevt.dll 1.50.1085.0000 192.06 KB (196,669 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation	iasuser.dll 5.00.2134.1 25.77 KB (26,384 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\ntevt.dll	c:\winnt\system32\iasuser.dll
psapi.dll 5.00.2134.1 28.27 KB (28,944 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation	iasnap.dll 5.00.2195.2104 58.77 KB (60,176 bytes) 6/20/2002 1:57:46 PM Microsoft Corporation
c:\winnt\system32\psapi.dll	c:\winnt\system32\iasnap.dll
framedyn.dll 1.50.1085.0000 164.05 KB (167,992 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation	iaspipe.dll 5.00.2134.1 41.77 KB (42,768 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\framedyn.dll	c:\winnt\system32\iaspipe.dll
cimwin32.dll 1.50.1085.0038 1.02 MB (1,073,232 bytes) 6/20/2002 1:58:13 PM Microsoft Corporation	expsrv.dll 6.0.8540 370.27 KB (379,152 bytes) 6/20/2002 1:57:45 PM Microsoft Corporation
c:\winnt\system32\wbem\cimwin32.dll	c:\winnt\system32\expsrv.dll
wbemsvc.dll 1.50.1085.0007 40.07 KB (41,036 bytes) 6/20/2002 1:58:14 PM Microsoft Corporation	vbajet32.dll 6.1.8268 30.27 KB (30,992 bytes) 6/20/2002 1:58:08 PM Microsoft Corporation
c:\winnt\system32\wbem\wbemsvc.dll	c:\winnt\system32\vbajet32.dll
wbemess.dll 1.50.1085.0039 364.07 KB (372,804 bytes) 6/20/2002 1:58:14 PM Microsoft Corporation	msjtes40.dll 4.00.4229.0 236.27 KB (241,936 bytes) 6/20/2002 1:57:55 PM Microsoft Corporation
c:\winnt\system32\wbem\wbemess.dll	c:\winnt\system32\msjtes40.dll
fastprox.dll 1.50.1085.0037 144.08 KB (147,536 bytes) 6/20/2002 1:58:13 PM Microsoft Corporation	oledb32r.dll 2.62.7926.0 68.27 KB (69,904 bytes) 6/20/2002 2:12:06 PM Microsoft Corporation
c:\winnt\system32\wbem\fastprox.dll	c:\program files\common files\system\ole db\oledb32r.dll
wbemcore.dll 1.50.1085.0036 628.07 KB (643,140 bytes) 6/20/2002 1:58:14 PM Microsoft Corporation	msdart.dll 2.62.7926.0 144.27 KB (147,728 bytes) 6/20/2002 2:12:05 PM Microsoft Corporation
c:\winnt\system32\wbem\wbemcore.dll	c:\winnt\system32\msdart.dll
wbemcomm.dll 1.50.1085.0021 692.07 KB (708,675 bytes) 6/20/2002 1:58:14 PM Microsoft Corporation	oledb32.dll 2.62.7926.0 448.27 KB (459,024 bytes) 6/20/2002 2:12:06 PM Microsoft Corporation
c:\winnt\system32\wbem\wbemcomm.dll	c:\program files\common files\system\ole db\oledb32.dll
winmgmt.exe 1.50.1085.0029 192.08 KB (196,685 bytes) 6/20/2002 1:58:14 PM Microsoft Corporation	msjint40.dll 4.00.2927.2 148.27 KB (151,824 bytes) 6/20/2002 1:57:55 PM Microsoft Corporation
c:\winnt\system32\wbem\winmgmt.exe	c:\winnt\system32\msjint40.dll
simptcp.dll 5.00.2134.1 19.27 KB (19,728 bytes) 6/20/2002 6:39:17 AM Microsoft Corporation	msjter40.dll 4.00.2927.2 52.27 KB (53,520 bytes) 6/20/2002 1:57:55 PM Microsoft Corporation
c:\winnt\system32\simptcp.dll	c:\winnt\system32\msjter40.dll
tcpsvcs.exe 5.00.2134.1 24.77 KB (25,360 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation	mswstr10.dll 4.00.3829.2 600.27 KB (614,672 bytes) 6/20/2002 1:57:57 PM Microsoft Corporation
c:\winnt\system32\tcpsvcs.exe	c:\winnt\system32\mswstr10.dll
ibmhpasv.exe 5.0.3.2 15.56 KB (15,935 bytes) 6/10/2002 7:22:05 PM IBM Corporation	msjet40.dll 4.00.4431.3 1.43 MB (1,503,504 bytes) 6/20/2002 1:57:55 PM Microsoft Corporation
c:\winnt\system32\ibmhpasv.exe	c:\winnt\system32\msjet40.dll
rasdlg.dll 5.00.2195.2671 514.27 KB (526,608 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation	msjtoledb40.dll 4.00.4331.4 340.27 KB (348,432 bytes) 6/20/2002 1:57:55 PM Microsoft Corporation
c:\winnt\system32\rasdlg.dll	c:\winnt\system32\msjtoledb40.dll
netcfgx.dll 5.00.2195.2228 534.77 KB (547,600 bytes) 6/20/2002 1:57:58 PM Microsoft Corporation	iasrad.dll 5.00.2139.1 94.27 KB (96,528 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation
c:\winnt\system32\netcfgx.dll	c:\winnt\system32\iasrad.dll
sens.dll 5.00.2163.1 36.77 KB (37,648 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation	iassam.dll 5.00.2160.1 96.27 KB (98,576 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation
c:\winnt\system32\sens.dll	c:\winnt\system32\iassam.dll
rasmans.dll 5.00.2195.2728 147.27 KB (150,800 bytes) 6/20/2002 1:58:03 PM Microsoft Corporation	iasads.dll 5.00.2134.1 73.77 KB (75,536 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation
c:\winnt\system32\rasmans.dll	c:\winnt\system32\iasads.dll
wmi.dll 5.00.2191.1 6.27 KB (6,416 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation	iaspolicy.dll 5.00.2134.1 25.27 KB (25,872 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation
c:\winnt\system32\wmi.dll	c:\winnt\system32\iaspolicy.dll
netshell.dll 5.00.2195.2779 457.27 KB (468,240 bytes) 6/20/2002 1:57:59 PM Microsoft Corporation	iassvcs.dll 5.00.2195.2104 58.77 KB (60,176 bytes) 6/20/2002 1:57:46 PM Microsoft Corporation
c:\winnt\system32\netshell.dll	c:\winnt\system32\iassvcs.dll
netman.dll 5.00.2195.2779 89.27 KB (91,408 bytes) 6/20/2002 1:57:59 PM Microsoft Corporation	iasdo.dll 5.00.2195.2104 261.77 KB (268,048 bytes) 6/20/2002 1:57:46 PM Microsoft Corporation
c:\winnt\system32\netman.dll	c:\winnt\system32\iasdo.dll

ias.dll	5.00.2134.1	7.27 KB (7,440 bytes)	12/7/1999 6:00:00 AM	ntdsatq.dll	5.00.2195.2878	31.27 KB (32,016 bytes)	6/20/2002 1:58:00 PM
Microsoft Corporation	c:\winnt\system32\ias.dll			Microsoft Corporation	c:\winnt\system32\ntdsatq.dll		
mtxoci.dll	2000.2.3471.1	101.77 KB (104,208 bytes)	6/20/2002 1:57:58 PM	ntdsa.dll	5.00.2195.2899	990.77 KB (1,014,544 bytes)	6/20/2002 1:57:59 PM
Microsoft Corporation	c:\winnt\system32\mtxoci.dll			Microsoft Corporation	c:\winnt\system32\ntdsa.dll		
resutils.dll	5.00.2195.2787	39.77 KB (40,720 bytes)	6/20/2002 1:58:03 PM	kdcsvc.dll	5.00.2195.2878	137.77 KB (141,072 bytes)	6/20/2002 1:57:49 PM
Microsoft Corporation	c:\winnt\system32\resutils.dll			Microsoft Corporation	c:\winnt\system32\kdcsvc.dll		
clusapi.dll	5.00.2195.2104	54.27 KB (55,568 bytes)	6/20/2002 1:57:41 PM	sfmapi.dll	5.00.2134.1	38.77 KB (39,696 bytes)	12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\clusapi.dll			Microsoft Corporation	c:\winnt\system32\sfmapi.dll		
msvcpx50.dll	5.00.7051	552.50 KB (565,760 bytes)	12/7/1999 6:00:00 AM	rassfm.dll	5.00.2195.2671	21.27 KB (21,776 bytes)	6/20/2002 1:58:03 PM
Microsoft Corporation	c:\winnt\system32\msvcpx50.dll			Microsoft Corporation	c:\winnt\system32\rassfm.dll		
xolehlp.dll	1999.9.3421.3	17.27 KB (17,680 bytes)	6/20/2002 6:39:24 AM	rsabase.dll	5.00.2195.2228	128.27 KB (131,344 bytes)	5/4/2001 12:05:02 PM
Microsoft Corporation	c:\winnt\system32\xolehlp.dll			Microsoft Corporation	c:\winnt\system32\rsabase.dll		
msdtclog.dll	1999.9.3421.3	89.77 KB (91,920 bytes)	6/20/2002 6:39:24 AM	schannel.dll	5.00.2195.2922	138.27 KB (141,584 bytes)	5/4/2001 12:05:02 PM
Microsoft Corporation	c:\winnt\system32\msdtclog.dll			Microsoft Corporation	c:\winnt\system32\schannel.dll		
mtxclu.dll	2000.2.3471.1	51.27 KB (52,496 bytes)	6/20/2002 1:57:58 PM	netlogon.dll	5.00.2195.2865	357.77 KB (366,352 bytes)	6/20/2002 1:57:58 PM
Microsoft Corporation	c:\winnt\system32\mtxclu.dll			Microsoft Corporation	c:\winnt\system32\netlogon.dll		
msdteprx.dll	2000.2.3471.1	665.77 KB (681,744 bytes)	6/20/2002 1:57:51 PM	kerberos.dll	5.00.2195.2913	198.77 KB (203,536 bytes)	6/20/2002 1:57:49 PM
Microsoft Corporation	c:\winnt\system32\msdteprx.dll			Microsoft Corporation	c:\winnt\system32\kerberos.dll		
txfaux.dll	2000.2.3471.1	374.27 KB (383,248 bytes)	6/20/2002 1:58:07 PM	msprivs.dll	5.00.2154.1	41.50 KB (42,496 bytes)	12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\txfaux.dll			Microsoft Corporation	c:\winnt\system32\msprivs.dll		
msdtctm.dll	2000.2.3471.1	1.07 MB (1,120,528 bytes)	6/20/2002 1:57:51 PM	samsrv.dll	5.00.2195.2918	369.77 KB (378,640 bytes)	12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\msdtctm.dll			Microsoft Corporation	c:\winnt\system32\samsrv.dll		
msdtc.exe	1999.9.3421.3	6.77 KB (6,928 bytes)	6/20/2002 6:39:24 AM	lsasrv.dll	5.00.2195.2964	492.77 KB (504,592 bytes)	12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\msdtc.exe			Microsoft Corporation	c:\winnt\system32\lsasrv.dll		
rasadhlp.dll	5.00.2168.1	7.27 KB (7,440 bytes)	12/7/1999 6:00:00 AM	lsass.exe	5.00.2195.2964	32.77 KB (33,552 bytes)	12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\rasadhlp.dll			Microsoft Corporation	c:\winnt\system32\lsass.exe		
winnr.dll	5.00.2160.1	18.77 KB (19,216 bytes)	12/7/1999 6:00:00 AM	ntlsapi.dll	5.00.2134.1	6.77 KB (6,928 bytes)	12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\winnr.dll			Microsoft Corporation	c:\winnt\system32\ntlsapi.dll		
wshtcpip.dll	5.00.2195.2104	17.27 KB (17,680 bytes)	6/20/2002 1:58:09 PM	msi.dll	1.11.2405.0	1.69 MB (1,767,184 bytes)	6/20/2002 1:57:53 PM
Microsoft Corporation	c:\winnt\system32\wshtcpip.dll			Microsoft Corporation	c:\winnt\system32\msi.dll		
msafd.dll	5.00.2195.2779	106.77 KB (109,328 bytes)	6/20/2002 1:57:51 PM	apmgmts.dll	5.00.2168.1	117.77 KB (120,592 bytes)	12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\msafd.dll			Microsoft Corporation	c:\winnt\system32\apmgmts.dll		
rpss.dll	5.00.2195.2815	231.27 KB (236,816 bytes)	6/20/2002 1:58:04 PM	dhcpcsvc.dll	5.00.2195.2778	88.77 KB (90,896 bytes)	12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\rpss.dll			Microsoft Corporation	c:\winnt\system32\dhcpcsvc.dll		
svchost.exe	5.00.2134.1	7.77 KB (7,952 bytes)	12/7/1999 6:00:00 AM	tapi32.dll	5.00.2182.1	123.27 KB (126,224 bytes)	12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\svchost.exe			Microsoft Corporation	c:\winnt\system32\tapi32.dll		
scecli.dll	5.00.2195.2780	105.27 KB (107,792 bytes)	6/20/2002 1:58:04 PM	rasman.dll	5.00.2195.2780	54.77 KB (56,080 bytes)	12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\scecli.dll			Microsoft Corporation	c:\winnt\system32\rasman.dll		
atl.dll	3.00.8449	57.56 KB (58,938 bytes)	12/7/1999 6:00:00 AM	rasapi32.dll	5.00.2195.2671	189.77 KB (194,320 bytes)	12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\atl.dll			Microsoft Corporation	c:\winnt\system32\rasapi32.dll		
certcli.dll	5.00.2195.2778	130.77 KB (133,904 bytes)	6/20/2002 1:57:41 PM	rtutils.dll	5.00.2168.1	43.77 KB (44,816 bytes)	12/7/1999 6:00:00 AM
Microsoft Corporation	c:\winnt\system32\certcli.dll			Microsoft Corporation	c:\winnt\system32\rtutils.dll		
esent.dll	6.0.3940.13	1.08 MB (1,135,376 bytes)	6/20/2002 1:57:45 PM	adslldpc.dll	5.00.2195.2842	127.27 KB (130,320 bytes)	6/20/2002 1:57:37 PM
Microsoft Corporation	c:\winnt\system32\esent.dll			Microsoft Corporation	c:\winnt\system32\adslldpc.dll		
mswsock.dll	5.00.2195.2871	62.77 KB (64,272 bytes)	6/20/2002 1:57:57 PM	activeds.dll	5.00.2195.2778	174.77 KB (178,960 bytes)	6/20/2002 1:57:32 PM
Microsoft Corporation	c:\winnt\system32\mswsock.dll			Microsoft Corporation	c:\winnt\system32\activeds.dll		

mprapi.dll	5.00.2181.1	79.27 KB (81,168 bytes)	12/7/1999	comdlg32.dll	5.00.3103.1000	236.77 KB (242,448 bytes)	
6:00:00 AM	Microsoft Corporation			12/7/1999 6:00:00 AM	Microsoft Corporation		
c:\winnt\system32\mprapi.dll				c:\winnt\system32\comdlg32.dll			
iphlpapi.dll	5.00.2173.2	67.77 KB (69,392 bytes)		netui2.dll	5.00.2134.1	280.27 KB (286,992 bytes)	12/7/1999
12/7/1999 6:00:00 AM	Microsoft Corporation			6:00:00 AM	Microsoft Corporation		
c:\winnt\system32\iphlpapi.dll				c:\winnt\system32\netui2.dll			
mr20.dll	5.00.2195.2871	35.77 KB (36,624 bytes)	6/20/2002	mprui.dll	5.00.2195.2104	54.77 KB (56,080 bytes)	6/20/2002
1:58:04 PM	Microsoft Corporation			1:57:51 PM	Microsoft Corporation		
c:\winnt\system32\mr20.dll				c:\winnt\system32\mprui.dll			
wmicore.dll	5.00.2195.2842	72.27 KB (74,000 bytes)		netui1.dll	5.00.2134.1	210.27 KB (215,312 bytes)	12/7/1999
6/20/2002 1:58:09 PM	Microsoft Corporation			6:00:00 AM	Microsoft Corporation		
c:\winnt\system32\wmicore.dll				c:\winnt\system32\netui1.dll			
psbase.dll	5.00.2195.2779	111.77 KB (114,448 bytes)	6/20/2002	netui0.dll	5.00.2134.1	70.27 KB (71,952 bytes)	12/7/1999
1:58:02 PM	Microsoft Corporation			6:00:00 AM	Microsoft Corporation		
c:\winnt\system32\psbase.dll				c:\winnt\system32\netui0.dll			
cryptsvc.dll	5.00.2181.1	61.77 KB (63,248 bytes)		ntlanman.dll	5.00.2157.1	35.27 KB (36,112 bytes)	
12/7/1999 6:00:00 AM	Microsoft Corporation			12/7/1999 6:00:00 AM	Microsoft Corporation		
c:\winnt\system32\cryptsvc.dll				c:\winnt\system32\ntlanman.dll			
cryptdll.dll	5.00.2135.1	41.27 KB (42,256 bytes)	12/7/1999	mpr.dll	5.00.2195.2779	53.27 KB (54,544 bytes)	6/20/2002
6:00:00 AM	Microsoft Corporation			1:57:50 PM	Microsoft Corporation		
c:\winnt\system32\cryptdll.dll				c:\winnt\system32\mpr.dll			
wkssvc.dll	5.00.2195.2780	95.27 KB (97,552 bytes)	12/7/1999	csui.dll	5.00.2195.2959	228.27 KB (233,744 bytes)	6/20/2002
6:00:00 AM	Microsoft Corporation			1:57:42 PM	Microsoft Corporation		
c:\winnt\system32\wkssvc.dll				c:\winnt\system32\csui.dll			
srsvcs.dll	5.00.2195.2904	79.27 KB (81,168 bytes)	12/7/1999	winspool.drv	5.00.2195.2780	109.77 KB (112,400 bytes)	
6:00:00 AM	Microsoft Corporation			12/7/1999 6:00:00 AM	Microsoft Corporation		
c:\winnt\system32\srsvcs.dll				c:\winnt\system32\winspool.drv			
cfgmgr32.dll	5.00.2134.1	16.77 KB (17,168 bytes)		winscard.dll	5.00.2134.1	77.27 KB (79,120 bytes)	
12/7/1999 6:00:00 AM	Microsoft Corporation			12/7/1999 6:00:00 AM	Microsoft Corporation		
c:\winnt\system32\cfgmgr32.dll				c:\winnt\system32\winscard.dll			
dmserver.dll	2195.2778.297.3	11.77 KB (12,048 bytes)		wlnotify.dll	5.00.2195.2780	53.77 KB (55,056 bytes)	
6/20/2002 1:57:43 PM	VERITAS Software Corp.			6/20/2002 1:58:09 PM	Microsoft Corporation		
c:\winnt\system32\dmserver.dll				c:\winnt\system32\wlnotify.dll			
winsta.dll	5.00.2195.2386	36.77 KB (37,648 bytes)	6/20/2002	csd.dll	5.00.2195.2401	98.27 KB (100,624 bytes)	6/20/2002
1:58:09 PM	Microsoft Corporation			1:57:42 PM	Microsoft Corporation		
c:\winnt\system32\winsta.dll				c:\winnt\system32\csd.dll			
icmp.dll	5.00.2134.1	7.27 KB (7,440 bytes)	12/7/1999 6:00:00 AM	lz32.dll	5.00.2134.1	9.77 KB (10,000 bytes)	12/7/1999
Microsoft Corporation				6:00:00 AM	Microsoft Corporation		
c:\winnt\system32\icmp.dll				c:\winnt\system32\lz32.dll			
lmhsvc.dll	5.00.2195.2778	9.77 KB (10,000 bytes)	12/7/1999	version.dll	5.00.2134.1	15.77 KB (16,144 bytes)	12/7/1999
6:00:00 AM	Microsoft Corporation			6:00:00 AM	Microsoft Corporation		
c:\winnt\system32\lmhsvc.dll				c:\winnt\system32\version.dll			
eventlog.dll	5.00.2178.1	43.77 KB (44,816 bytes)		rsaenh.dll	5.00.2195.2228	130.77 KB (133,904 bytes)	6/20/2002
12/7/1999 6:00:00 AM	Microsoft Corporation			1:58:54 PM	Microsoft Corporation		
c:\winnt\system32\eventlog.dll				c:\winnt\system32\rsaenh.dll			
ntdsapi.dll	5.00.2195.2661	55.77 KB (57,104 bytes)	6/20/2002	mcat32.dll	5.131.2134.1	7.77 KB (7,952 bytes)	12/7/1999
1:57:59 PM	Microsoft Corporation			6:00:00 AM	Microsoft Corporation		
c:\winnt\system32\ntdsapi.dll				c:\winnt\system32\mcat32.dll			
scsersv.dll	5.00.2195.2780	226.27 KB (231,696 bytes)	6/20/2002	ole32.dll	5.00.2195.2887	969.77 KB (993,040 bytes)	6/20/2002
1:58:04 PM	Microsoft Corporation			1:58:01 PM	Microsoft Corporation		
c:\winnt\system32\scsersv.dll				c:\winnt\system32\ole32.dll			
umpnprmgr.dll	5.00.2182.1	86.27 KB (88,336 bytes)		imagehlp.dll	5.00.2195.2778	125.77 KB (128,784 bytes)	
12/7/1999 6:00:00 AM	Microsoft Corporation			5/4/2001 12:05:02 PM	Microsoft Corporation		
c:\winnt\system32\umpnprmgr.dll				c:\winnt\system32\imagehlp.dll			
services.exe	5.00.2195.2780	86.77 KB (88,848 bytes)		msasn1.dll	5.00.2134.1	51.27 KB (52,496 bytes)	12/7/1999
12/7/1999 6:00:00 AM	Microsoft Corporation			6:00:00 AM	Microsoft Corporation		
c:\winnt\system32\services.exe				c:\winnt\system32\msasn1.dll			
msv1_0.dll	5.00.2195.2900	111.77 KB (114,448 bytes)	12/7/1999	crypt32.dll	5.131.2195.2833	451.27 KB (462,096 bytes)	6/20/2002
6:00:00 AM	Microsoft Corporation			1:57:42 PM	Microsoft Corporation		
c:\winnt\system32\msv1_0.dll				c:\winnt\system32\crypt32.dll			
clbcatq.dll	2000.2.3471.1	496.77 KB (508,688 bytes)	6/20/2002	wintrust.dll	5.131.2195.2779	162.27 KB (166,160 bytes)	
1:57:41 PM	Microsoft Corporation			6/20/2002 1:58:09 PM	Microsoft Corporation		
c:\winnt\system32\clbcatq.dll				c:\winnt\system32\wintrust.dll			
oleaut32.dll	2.40.4517	612.27 KB (626,960 bytes)	12/7/1999	setupapi.dll	5.00.2195.2663	555.77 KB (569,104 bytes)	
6:00:00 AM	Microsoft Corporation			12/7/1999 6:00:00 AM	Microsoft Corporation		
c:\winnt\system32\oleaut32.dll				c:\winnt\system32\setupapi.dll			
netmsg.dll	5.00.2137.1	152.50 KB (156,160 bytes)	12/7/1999				
6:00:00 AM	Microsoft Corporation						
c:\winnt\system32\netmsg.dll							

winmm.dll 5.00.2161.1	184.77 KB (189,200 bytes)	12/7/1999	kernel32.dll 5.00.2195.2778	714.77 KB (731,920 bytes)	
6:00:00 AM Microsoft Corporation			12/7/1999 6:00:00 AM Microsoft Corporation		
c:\winnt\system32\winmm.dll			c:\winnt\system32\kernel32.dll		
comctl32.dll 5.81	537.77 KB (550,672 bytes)	12/7/1999	msvcrt.dll 6.10.8924.0	284.05 KB (290,869 bytes)	5/4/2001
6:00:00 AM Microsoft Corporation			12:05:02 PM Microsoft Corporation		
c:\winnt\system32\comctl32.dll			c:\winnt\system32\msvcrt.dll		
shlwapi.dll 5.00.3315.1000	282.77 KB (289,552 bytes)	6/20/2002	winlogon.exe 5.00.2195.2953	173.77 KB (177,936 bytes)	
1:58:06 PM Microsoft Corporation			12/7/1999 6:00:00 AM Microsoft Corporation		
c:\winnt\system32\shlwapi.dll			c:\winnt\system32\winlogon.exe		
shell32.dll 5.00.3315.2902	2.25 MB (2,359,056 bytes)	6/20/2002	sfcdll.dll 5.00.2195.2967	948.27 KB (971,024 bytes)	6/20/2002
1:58:05 PM Microsoft Corporation			1:58:05 PM Microsoft Corporation		
c:\winnt\system32\shell32.dll			c:\winnt\system32\sfcdll.dll		
msgina.dll 5.00.2195.2779	324.27 KB (332,048 bytes)	12/7/1999	ntdll.dll 5.00.2195.2779	478.77 KB (490,256 bytes)	5/4/2001
6:00:00 AM Microsoft Corporation			12:05:02 PM Microsoft Corporation		
c:\winnt\system32\msgina.dll			c:\winnt\system32\ntdll.dll		
wsock32.dll 5.00.2195.2871	21.27 KB (21,776 bytes)		smss.exe 5.00.2195.2901	44.27 KB (45,328 bytes)	12/7/1999
6/20/2002 1:58:10 PM Microsoft Corporation			6:00:00 AM Microsoft Corporation		
c:\winnt\system32\wsock32.dll			c:\winnt\system32\smss.exe		
dnsapi.dll 5.00.2195.2785	130.77 KB (133,904 bytes)	6/20/2002	[Services]		
1:57:43 PM Microsoft Corporation			Display Name	Name	State
c:\winnt\system32\dnsapi.dll			Path	Error Control	Start Name
wldap32.dll 5.00.2195.2797	125.27 KB (128,272 bytes)		Alerter	Alerter	Stopped
6/20/2002 1:58:09 PM Microsoft Corporation			c:\winnt\system32\services.exe	Normal	LocalSystem
c:\winnt\system32\wldap32.dll			Application Management	AppMgmt	Running
ws2help.dll 5.00.2134.1	17.77 KB (18,192 bytes)		Process c:\winnt\system32\services.exe	Normal	LocalSystem
12/7/1999 6:00:00 AM Microsoft Corporation			Computer Browser	Browser	Stopped
c:\winnt\system32\ws2help.dll			c:\winnt\system32\services.exe	Normal	LocalSystem
ws2_32.dll 5.00.2195.2780	67.77 KB (69,392 bytes)	6/20/2002	Indexing Service	cisvc	Stopped
1:58:09 PM Microsoft Corporation			c:\winnt\system32\cisvc.exe	Normal	LocalSystem
c:\winnt\system32\ws2_32.dll			ClipBook	ClipSrv	Stopped
samlib.dll 5.00.2195.2780	49.77 KB (50,960 bytes)	12/7/1999	c:\winnt\system32\clipsrv.exe	Normal	LocalSystem
6:00:00 AM Microsoft Corporation			Distributed File System	Dfs	Stopped
c:\winnt\system32\samlib.dll			Process c:\winnt\system32\dfssvc.exe	Normal	LocalSystem
netrap.dll 5.00.2134.1	11.27 KB (11,536 bytes)	12/7/1999	DHCP Client	Dhcp	Stopped
6:00:00 AM Microsoft Corporation			c:\winnt\system32\services.exe	Normal	LocalSystem
c:\winnt\system32\netrap.dll			Logical Disk Manager Administrative Service	dmadmin	Stopped
netapi32.dll 5.00.2195.2808	303.77 KB (311,056 bytes)		Manual Share Process	c:\winnt\system32\dmadmin.exe /com	
6/20/2002 1:57:58 PM Microsoft Corporation			Normal LocalSystem	0	
c:\winnt\system32\netapi32.dll			Logical Disk Manager dmserver	Running	Auto
profmap.dll 5.00.2181.1	29.27 KB (29,968 bytes)		c:\winnt\system32\services.exe	Normal	LocalSystem
12/7/1999 6:00:00 AM Microsoft Corporation			DNS Client	Dnscache	Stopped
c:\winnt\system32\profmap.dll			c:\winnt\system32\services.exe	Normal	LocalSystem
secur32.dll 5.00.2195.2862	46.77 KB (47,888 bytes)	6/20/2002	Event Log	Eventlog	Running
1:58:04 PM Microsoft Corporation			c:\winnt\system32\services.exe	Normal	LocalSystem
c:\winnt\system32\secur32.dll			COM+ Event System	EventSystem	Running
sfcdll.dll 5.00.2195.2896	92.11 KB (94,320 bytes)	6/20/2002	Process c:\winnt\system32\svchost.exe -k netsvcs	Normal	LocalSystem
1:58:05 PM Microsoft Corporation			Fax Service	Fax	Stopped
c:\winnt\system32\sfcdll.dll			c:\winnt\system32\faxsvc.exe	Normal	LocalSystem
nddeapi.dll 5.00.2137.1	15.27 KB (15,632 bytes)	12/7/1999	Mylex Global Array Manager Server	gamscm	Running
6:00:00 AM Microsoft Corporation			Own Process	c:\winnt\system32\gamssrv\gamscm.exe	Normal
c:\winnt\system32\nddeapi.dll			LocalSystem	0	
userenv.dll 5.00.2195.2780	361.77 KB (370,448 bytes)	12/7/1999	Internet Authentication Service	IAS	Running
6:00:00 AM Microsoft Corporation			Process c:\winnt\system32\svchost.exe -k netsvcs	Normal	LocalSystem
c:\winnt\system32\userenv.dll			LocalSystem	0	
user32.dll 5.00.2195.2821	392.77 KB (402,192 bytes)	12/7/1999	IBM Active PCI Alert Service	IBMHPAS	Running
6:00:00 AM Microsoft Corporation			Process c:\winnt\system32\ibmhpasv.exe	Normal	LocalSystem
c:\winnt\system32\user32.dll			IIS Admin Service	IISADMIN	Running
gdi32.dll 5.00.2195.2778	228.77 KB (234,256 bytes)	12/7/1999	Process c:\winnt\system32\inetrv\inetinfo.exe	Normal	LocalSystem
6:00:00 AM Microsoft Corporation			LocalSystem	0	
c:\winnt\system32\gdi32.dll			Intersite Messaging	IsmServ	Stopped
rpert4.dll 5.00.2195.2832	437.27 KB (447,760 bytes)	6/20/2002	c:\winnt\system32\ismssrv.exe	Normal	LocalSystem
1:58:04 PM Microsoft Corporation			Kerberos Key Distribution Center	kdc	Stopped
c:\winnt\system32\rpert4.dll			Share Process	c:\winnt\system32\lsass.exe	Normal
advapi32.dll 5.00.2195.2867	351.77 KB (360,208 bytes)		LocalSystem	0	
12/7/1999 6:00:00 AM Microsoft Corporation					
c:\winnt\system32\advapi32.dll					

Server	lanmanserver	Running	Auto	Share Process		Smart Card Helper	SCardDrv	Stopped	Manual	Share Process	
c:\winnt\system32\services.exe		Normal	LocalSystem	0		c:\winnt\system32\scardsvr.exe		Ignore	LocalSystem	0	
Workstation	lanmanworkstation	Running	Auto	Share		Smart Card	SCardSvr	Stopped	Manual	Share Process	
Process	c:\winnt\system32\services.exe	Normal	LocalSystem	0		c:\winnt\system32\scardsvr.exe		Ignore	LocalSystem	0	
License Logging Service	LicenseService	Stopped	Manual			Task Scheduler	Schedule	Stopped	Manual	Share Process	
Own Process	c:\winnt\system32\llssrv.exe	Normal				c:\winnt\system32\mstask.exe		Normal	LocalSystem	0	
LocalSystem	0					RunAs Service	seclogon	Stopped	Manual	Share Process	
TCP/IP NetBIOS Helper Service	LmHosts	Running	Auto	Share		c:\winnt\system32\services.exe		Ignore	LocalSystem	0	
Process	c:\winnt\system32\services.exe	Normal	LocalSystem	0		System Event Notification	SENS	Stopped	Manual	Share	
Messenger Messenger	Stopped	Manual	Share Process			Process	c:\winnt\system32\svchost.exe -k netsvcs	Normal			
c:\winnt\system32\services.exe		Normal	LocalSystem	0		LocalSystem	0				
NetMeeting Remote Desktop Sharing	mnmsrvc	Stopped	Manual			Internet Connection Sharing		SharedAccess	Stopped	Manual	
Own Process	c:\winnt\system32\mnmsrvc.exe	Normal				Share Process	c:\winnt\system32\svchost.exe -k netsvcs	Normal			
LocalSystem	0					LocalSystem	0				
Distributed Transaction Coordinator	MSDTC	Running	Auto			Simple TCP/IP Services	SimpTcp	Running	Auto	Share	
Own Process	c:\winnt\system32\msdtc.exe	Normal				Process	c:\winnt\system32\tcpvcs.exe	Normal	LocalSystem	0	
LocalSystem	0					Print Spooler	Spooler	Stopped	Manual	Own Process	
Windows Installer	MSIServer	Stopped	Manual	Share Process		c:\winnt\system32\spoolsv.exe		Normal	LocalSystem	0	
c:\winnt\system32\msiexec.exe /v		Normal	LocalSystem	0		SQLSERVERAGENT	SQLSERVERAGENT	Stopped			
MSSQLSERVER	MSSQLSERVER	Stopped	Manual	Own		Manual	Own Process	c:\progra~1\microso~3\mssql\bin\sqlagent.exe			
Process	c:\progra~1\microso~3\mssql\bin\sqlservr.exe	Normal				Normal	LocalSystem	0			
LocalSystem	0					Performance Logs and Alerts	SysmonLog	Stopped	Manual		
MSSQLServerADHelper	MSSQLServerADHelper	Stopped				Own Process	c:\winnt\system32\smlogsvc.exe	Normal			
Manual	Own Process	c:\program files\microsoft sql				LocalSystem	0				
server\80\tools\bin\sqladhlp.exe		Normal	LocalSystem	0		Telephony TapiSrv	Running	Manual	Share Process		
Network DDE	NetDDE	Stopped	Manual	Share Process		c:\winnt\system32\svchost.exe -k tapisrv		Normal	LocalSystem	0	
c:\winnt\system32\netdde.exe		Normal	LocalSystem	0		Terminal Services	TermService	Stopped	Disabled	Own	
Network DDE DSDMNetDDEdsdm		Stopped	Manual	Share		Process	c:\winnt\system32\termsrv.exe	Normal	LocalSystem	0	
Process	c:\winnt\system32\netdde.exe	Normal	LocalSystem	0		Telnet	TlntSvr	Stopped	Manual	Own Process	
Net Logon Netlogon	Stopped	Manual	Share Process			c:\winnt\system32\tlntsvr.exe		Normal	LocalSystem	0	
c:\winnt\system32\lsass.exe		Normal	LocalSystem	0		Distributed Link Tracking Server	TrkSvr	Stopped	Manual	Share	
Network Connections Netman	Running	Manual	Share Process			Process	c:\winnt\system32\services.exe	Normal	LocalSystem	0	
c:\winnt\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		Distributed Link Tracking Client	TrkWks	Stopped	Manual	Share	
File Replication	NtFrs	Stopped	Manual	Own Process		Process	c:\winnt\system32\services.exe	Normal	LocalSystem	0	
c:\winnt\system32\ntfrs.exe		Ignore	LocalSystem	0		Uninterruptible Power Supply	UPS	Stopped	Manual	Own	
NT LM Security Support Provider	NtLmSsp	Stopped	Manual			Process	c:\winnt\system32\ups.exe	Normal	LocalSystem	0	
Share Process	c:\winnt\system32\lsass.exe	Normal				Utility Manager	UtilMan	Stopped	Manual	Own Process	
LocalSystem	0					c:\winnt\system32\utilman.exe		Normal	LocalSystem	0	
Removable Storage	NtmsSvc	Stopped	Manual	Share Process		Windows Time	W32Time	Stopped	Manual	Share Process	
c:\winnt\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		c:\winnt\system32\services.exe		Normal	LocalSystem	0	
Plug and Play	PlugPlay	Running	Auto	Share Process		World Wide Web Publishing Service	W3SVC	Running	Auto		
c:\winnt\system32\services.exe		Normal	LocalSystem	0		Share Process	c:\winnt\system32\inetrv\inetinfo.exe	Normal			
IPSEC Policy Agent	PolicyAgent	Stopped	Manual	Share		LocalSystem	0				
Process	c:\winnt\system32\lsass.exe	Normal	LocalSystem	0		Windows Management Instrumentation	WinMgmt	Running	Auto		
Protected Storage	ProtectedStorage	Running	Auto	Share		Own Process	c:\winnt\system32\wbem\winmgmt.exe	Ignore			
Process	c:\winnt\system32\services.exe	Normal	LocalSystem	0		LocalSystem	0				
Remote Access Auto Connection Manager	RasAuto	Stopped	Manual			Windows Management Instrumentation Driver Extensions	Wmi				
Share Process	c:\winnt\system32\svchost.exe -k netsvcs	Normal				Running	Manual	Share Process	c:\winnt\system32\services.exe		
LocalSystem	0					Normal	LocalSystem	0			
Remote Access Connection Manager	RasMan	Stopped	Manual			[Program Groups]					
Share Process	c:\winnt\system32\svchost.exe -k netsvcs	Normal				Group Name	Name	User Name			
LocalSystem	0					Accessories	Default User:Accessories	Default User			
Routing and Remote Access	RemoteAccess	Stopped	Disabled			Accessories\Accessibility		Default User:Accessories\Accessibility			
Share Process	c:\winnt\system32\svchost.exe -k netsvcs	Normal				Default User					
LocalSystem	0					Accessories\Entertainment		Default User:Accessories\Entertainment			
Remote Registry Service	RemoteRegistry	Stopped	Manual			Default User					
Own Process	c:\winnt\system32\regsvc.exe	Normal				Accessories\System Tools		Default User:Accessories\System Tools			
LocalSystem	0					Default User					
Remote Procedure Call (RPC) Locator	RpcLocator	Stopped				Startup	Default User:Startup	Default User			
Manual	Own Process	c:\winnt\system32\locator.exe	Normal			Accessories	All Users:Accessories	All Users			
LocalSystem	0					Accessories\Accessibility		All Users:Accessories\Accessibility			
Remote Procedure Call (RPC)	RpcSs	Running	Auto	Share		All Users					
Process	c:\winnt\system32\svchost -k rpcss	Normal				Accessories\Communications		All Users:Accessories\Communications			
LocalSystem	0					All Users					
QoS Admission Control (RSVP)	RSVP	Running	Auto	Own		Accessories\Entertainment		All Users:Accessories\Entertainment			
Process	c:\winnt\system32\rsvp.exe -s	Normal	LocalSystem	0		All Users					
Security Accounts Manager	SamSs	Running	Auto	Share							
Process	c:\winnt\system32\lsass.exe	Normal	LocalSystem	0							

Accessories\Games All Users:Accessories\Games All Users
 Accessories\Microsoft Script Debugger All Users:Accessories\Microsoft
 Script Debugger All Users
 Accessories\System Tools All Users:Accessories\System Tools
 All Users
 Administrative Tools All Users:Administrative Tools All Users
 Microsoft SQL ServerAll Users:Microsoft SQL Server All Users
 Microsoft SQL Server - Switch All Users:Microsoft SQL Server - Switch
 All Users
 Startup All Users:Startup All Users
 Accessories TPCC\Administrator:Accessories TPCC\Administrator
 Accessories\Accessibility
 TPCC\Administrator:Accessories\Accessibility TPCC\Administrator
 Accessories\Entertainment
 TPCC\Administrator:Accessories\Entertainment TPCC\Administrator
 Accessories\System Tools TPCC\Administrator:Accessories\System
 Tools TPCC\Administrator
 Startup TPCC\Administrator:Startup TPCC\Administrator

[Startup Programs]

Program Command User NameLocation
 No startup program information

[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	
Image Document	"C:\Program Files\Windows NT\Accessories\ImageVue\KodakImg.exe"	
WordPad Document	"%ProgramFiles%\Windows NT\Accessories\WORDPAD.EXE"	
Windows Media Services DRM Storage object		Not Available
Bitmap Image	mspaint.exe	

[Internet Explorer 5]

[Following are sub-categories of this main category]

[Summary]

Item	Value
Version	5.00.3315.1000
Build	53315.1000
Product ID	51879-270-5689806-05259
Application Path	C:\Program Files\Internet Explorer
Language	English (United States)
Active Printer	Not Available
Cipher Strength	168-bit
Content Advisor	Disabled
IEAK Install	No

[File Versions]

File	Version	Size	Date	Path	Company
advapi32.dll	5.0.2195.2867	352 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
advpack.dll	5.0.3103.1000	87 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
browseui.dll	5.0.3315.2846	35 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				

browseui.dll	5.0.3315.2846	789 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
ckenv.exe	5.0.2189.1	9 KB	12/7/1999 7:00:00 AM		
C:\WINNT\system32	Microsoft Corporation				
comctl32.dll	5.81.3103.1000	538 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
crypt32.dll	5.131.2195.2833	451 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
enhsg.dll	<File Missing>	Not Available	Not Available		
Not Available	Not Available				
iemigrat.dll	<File Missing>	Not Available	Not		
Available	Not Available	Not Available			
iesetup.dll	5.0.3103.1000	57 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
iexplore.exe	5.0.2920.0	59 KB	12/7/1999 7:00:00 AM		
C:\Program Files\Internet Explorer	Microsoft Corporation				
imagehlp.dll	5.0.2195.2778	126 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
imgghelp.dll	<File Missing>	Not Available	Not		
Available	Not Available	Not Available			
inseng.dll	5.0.3103.1000	72 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
jobexec.dll	5.0.0.1	47 KB	12/7/1999 7:00:00 AM		
C:\WINNT\system32	Microsoft Corporation				
jscrip.dll	5.1.0.5907	476 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
jsproxy.dll	5.0.2920.0	13 KB	12/7/1999 7:00:00 AM		
C:\WINNT\system32	Microsoft Corporation				
msaahtml.dll	<File Missing>	Not Available	Not		
Available	Not Available	Not Available			
mshtml.dll	5.0.3315.2870	2290 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
msjava.dll	5.0.3802.0	923 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
msoss.dll	<File Missing>	Not Available	Not Available		
Not Available	Not Available				
msxml.dll	8.0.5718.1	493 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
occache.dll	5.0.3103.1000	86 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
ole32.dll	5.0.2195.2887	970 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
oleaut32.dll	2.40.4517.0	612 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
olepro32.dll	5.0.4517.0	160 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
rsabase.dll	5.0.2195.2228	128 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
rsaenh.dll	5.0.2195.2228	131 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
rsapi32.dll	<File Missing>	Not Available	Not Available		
Not Available	Not Available				
rsasig.dll	<File Missing>	Not Available	Not Available		
Not Available	Not Available				
schannel.dll	5.1.2195.0	138 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
shdoc401.dll	<File Missing>	Not Available	Not		
Available	Not Available	Not Available			
shdocvw.dll	5.0.3315.2879	1078 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
shell32.dll	5.0.3315.2902	2304 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
shlwapi.dll	5.0.3315.1000	283 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				
url.dll	5.0.2920.0	82 KB	12/7/1999 7:00:00 AM		
C:\WINNT\system32	Microsoft Corporation				
urlmon.dll	5.0.3315.1000	441 KB	5/4/2001 12:05:02 PM		
C:\WINNT\system32	Microsoft Corporation				

vbscript.dll 5.1.0.5907 428 KB 5/4/2001 12:05:02 PM
 C:\WINNT\system32 Microsoft Corporation
 webcheck.dll 5.0.3315.1000 252 KB 5/4/2001 12:05:02 PM
 C:\WINNT\system32 Microsoft Corporation
 win.com 5.0.2134.1 24 KB 12/7/1999 7:00:00 AM
 C:\WINNT\system32 Microsoft Corporation
 wininet.dll 5.0.3315.1000 457 KB 5/4/2001 12:05:02 PM
 C:\WINNT\system32 Microsoft Corporation
 winsock.dll 3.10.0.103 3 KB 12/7/1999 7:00:00 AM
 C:\WINNT\system32 Microsoft Corporation
 wintrust.dll 5.131.2195.2779 162 KB 5/4/2001 12:05:02 PM
 C:\WINNT\system32 Microsoft Corporation
 wsock.vxd <File Missing> Not Available Not Available
 Not Available Not Available
 wsock32.dll 5.0.2195.2871 21 KB 5/4/2001 12:05:02 PM
 C:\WINNT\system32 Microsoft Corporation
 wsock32n.dll <File Missing> Not Available Not Available
 Available Not Available Not Available

[Connectivity]

Item	Value
Connection Preference	Never dial
EnableHttp1.1	1
ProxyHttp1.1	0

LAN Settings

AutoConfigProxy	wininet.dll
AutoProxyDetectMode	Enabled
AutoConfigURL	
Proxy	Disabled
ProxyServer	
ProxyOverride	

[Cache]

[Following are sub-categories of this main category]

[Summary]

Item	Value
Page Refresh Type	Automatic
Temporary Internet Files Folder	C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files
Total Disk Space	8667 MB
Available Disk Space	3474 MB
Maximum Cache Size	270 MB
Available Cache Size	271 MB

[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 other people certificate information available			

Administrator	Administrator	6/20/2002 to 5/27/2102
sha1RSA		

[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
Local intranet	Medium-low
Trusted sites	Low
Internet	Medium
Restricted sites	High

Disk Controller Configuration Parameters

Mylex eXtremeRAID 2000 Adapter 1

Begin

BeginGroup

PhysicalDevice0 = Channel=0, Target=0, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice1 = Channel=0, Target=1, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice2 = Channel=0, Target=2, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice3 = Channel=0, Target=3, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice4 = Channel=0, Target=4, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice5 = Channel=0, Target=5, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice6 = Channel=0, Target=6, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice7 = Channel=0, Target=8, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice8 = Channel=0, Target=9, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice9 = Channel=0, Target=10, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice10 = Channel=0, Target=11, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice11 = Channel=0, Target=12, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice12 = Channel=0, Target=13, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice13 = Channel=0, Target=14, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice14 = Channel=1, Target=0, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice15 = Channel=1, Target=1, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice16 = Channel=1, Target=2, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice17 = Channel=1, Target=3, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice18 = Channel=1, Target=4, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice19 = Channel=1, Target=5, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice20 = Channel=1, Target=6, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice21 = Channel=1, Target=8, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice22 = Channel=1, Target=9, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice23 = Channel=1, Target=10, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice24 = Channel=1, Target=11, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice25 = Channel=1, Target=12, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice26 = Channel=1, Target=13, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice27 = Channel=1, Target=14, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice28 = Channel=2, Target=0, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice29 = Channel=2, Target=1, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice30 = Channel=2, Target=2, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice31 = Channel=2, Target=3, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice32 = Channel=2, Target=4, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice33 = Channel=2, Target=5, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice34 = Channel=2, Target=6, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice35 = Channel=2, Target=8, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice36 = Channel=2, Target=9, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice37 = Channel=2, Target=10, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice38 = Channel=2, Target=11, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice39 = Channel=2, Target=12, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice40 = Channel=2, Target=13, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice41 = Channel=2, Target=14, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

IntermediateDevice0 = StripeSize=64kb, Raid=0, WriteThrough=1,
Size=242704mb,

(PhysicalDevice0, StartAddress=0mb, Size=17336mb),
(PhysicalDevice1, StartAddress=0mb, Size=17336mb),
(PhysicalDevice2, StartAddress=0mb, Size=17336mb),
(PhysicalDevice3, StartAddress=0mb, Size=17336mb),
(PhysicalDevice4, StartAddress=0mb, Size=17336mb),
(PhysicalDevice5, StartAddress=0mb, Size=17336mb),
(PhysicalDevice6, StartAddress=0mb, Size=17336mb),
(PhysicalDevice7, StartAddress=0mb, Size=17336mb),
(PhysicalDevice8, StartAddress=0mb, Size=17336mb),
(PhysicalDevice9, StartAddress=0mb, Size=17336mb),
(PhysicalDevice10, StartAddress=0mb, Size=17336mb),
(PhysicalDevice11, StartAddress=0mb, Size=17336mb),
(PhysicalDevice12, StartAddress=0mb, Size=17336mb),
(PhysicalDevice13, StartAddress=0mb, Size=17336mb);

IntermediateDevice1 = StripeSize=64kb, Raid=0, WriteThrough=1,
Size=242704mb,

(PhysicalDevice14, StartAddress=0mb, Size=17336mb),
(PhysicalDevice15, StartAddress=0mb, Size=17336mb),
(PhysicalDevice16, StartAddress=0mb, Size=17336mb),
(PhysicalDevice17, StartAddress=0mb, Size=17336mb),
(PhysicalDevice18, StartAddress=0mb, Size=17336mb),
(PhysicalDevice19, StartAddress=0mb, Size=17336mb),
(PhysicalDevice20, StartAddress=0mb, Size=17336mb),
(PhysicalDevice21, StartAddress=0mb, Size=17336mb),
(PhysicalDevice22, StartAddress=0mb, Size=17336mb),
(PhysicalDevice23, StartAddress=0mb, Size=17336mb),
(PhysicalDevice24, StartAddress=0mb, Size=17336mb),

```

        (PhysicalDevice25, StartAddress=0mb, Size=17336mb),
        (PhysicalDevice26, StartAddress=0mb, Size=17336mb),
        (PhysicalDevice27, StartAddress=0mb, Size=17336mb);

    IntermediateDevice2 = StripeSize=64kb, Raid=0, WriteThrough=1,
    Size=242704mb,

        (PhysicalDevice28, StartAddress=0mb, Size=17336mb),
        (PhysicalDevice29, StartAddress=0mb, Size=17336mb),
        (PhysicalDevice30, StartAddress=0mb, Size=17336mb),
        (PhysicalDevice31, StartAddress=0mb, Size=17336mb),
        (PhysicalDevice32, StartAddress=0mb, Size=17336mb),
        (PhysicalDevice33, StartAddress=0mb, Size=17336mb),
        (PhysicalDevice34, StartAddress=0mb, Size=17336mb),
        (PhysicalDevice35, StartAddress=0mb, Size=17336mb),
        (PhysicalDevice36, StartAddress=0mb, Size=17336mb),
        (PhysicalDevice37, StartAddress=0mb, Size=17336mb),
        (PhysicalDevice38, StartAddress=0mb, Size=17336mb),
        (PhysicalDevice39, StartAddress=0mb, Size=17336mb),
        (PhysicalDevice40, StartAddress=0mb, Size=17336mb),
        (PhysicalDevice41, StartAddress=0mb, Size=17336mb);

    LogicalDevice0 = StripeSize=64kb, Raid=12, WriteThrough=1,
    Size=728112mb, BIOSGeometry=8GB,

        (IntermediateDevice0, StartAddress=0mb,
    Size=242704mb),

        (IntermediateDevice1, StartAddress=0mb,
    Size=242704mb),

        (IntermediateDevice2, StartAddress=0mb,
    Size=242704mb);

EndGroup

BeginControllerParameter
    ControllerName = eXtremeRAID 2000;

    ControllerType = 28;

    FirmwareVersion = 7.01;

    CacheLineSize = 8KB;

    BackgroundTaskRate = 50;

    InitiatorID = 7;

    DiskStartupMode = AutoSpin;

```

```

    DevicesPerSpin = 2;

    InitialDelay = 6S;

    SequentialDelay = 0S;

    EnableDriveSizing = 0;

    EnableClustering = 0;

    EnableBGInit = 1;

    EnableReadAhead = 0;

    EnableBiosLoadDelay = 0;

    EnableForcedUnitAccess = 0;

    DisableBios = 1;

    EnableCDROMBoot = 0;

    EnableStorageWorks = 0;

    EnableSAFTE = 0;

    EnableSES = 0;

    EnableARM = 1;

    EnableOFM = 1;

    OEMCode = 0;

    StartupOption = 0;

EndControllerParameter

End

```

Mylex eXtremeRAID 2000 Adapter 2

Begin

BeginGroup

```

        PhysicalDevice0 = Channel=0, Target=0, Size=17340mb,
    State=Online,

```

```

        TransferSpeed=80MHz, TransferWidth=16Bit,
    MaxTag=16;

```

```

        PhysicalDevice1 = Channel=0, Target=1, Size=17340mb,
    State=Online,

```

```

        TransferSpeed=80MHz, TransferWidth=16Bit,
    MaxTag=16;

```

```

        PhysicalDevice2 = Channel=0, Target=2, Size=17340mb,
    State=Online,

```

```

        TransferSpeed=80MHz, TransferWidth=16Bit,
    MaxTag=16;

```

```

        PhysicalDevice3 = Channel=0, Target=3, Size=17340mb,
    State=Online,

```

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice4 = Channel=0, Target=4, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice5 = Channel=0, Target=5, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice6 = Channel=0, Target=6, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice7 = Channel=0, Target=8, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice8 = Channel=0, Target=9, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice9 = Channel=0, Target=10, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice10 = Channel=0, Target=11, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice11 = Channel=0, Target=12, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice12 = Channel=0, Target=13, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice13 = Channel=0, Target=14, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice14 = Channel=1, Target=0, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice15 = Channel=1, Target=1, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice16 = Channel=1, Target=2, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice17 = Channel=1, Target=3, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice18 = Channel=1, Target=4, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice19 = Channel=1, Target=5, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice20 = Channel=1, Target=6, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice21 = Channel=1, Target=8, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice22 = Channel=1, Target=9, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice23 = Channel=1, Target=10, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice24 = Channel=1, Target=11, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice25 = Channel=1, Target=12, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice26 = Channel=1, Target=13, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice27 = Channel=1, Target=14, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice28 = Channel=2, Target=0, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice29 = Channel=2, Target=1, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice30 = Channel=2, Target=2, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice31 = Channel=2, Target=3, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice32 = Channel=2, Target=4, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice33 = Channel=2, Target=5, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice34 = Channel=2, Target=6, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice35 = Channel=2, Target=8, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice36 = Channel=2, Target=9, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice37 = Channel=2, Target=10, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice38 = Channel=2, Target=11, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice39 = Channel=2, Target=12, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice40 = Channel=2, Target=13, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice41 = Channel=2, Target=14, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
IntermediateDevice0 = StripeSize=64kb, Raid=0, WriteThrough=1,
Size=242704mb,
(PhysicalDevice0, StartAddress=0mb, Size=17336mb),
(PhysicalDevice1, StartAddress=0mb, Size=17336mb),
(PhysicalDevice2, StartAddress=0mb, Size=17336mb),
(PhysicalDevice3, StartAddress=0mb, Size=17336mb),
(PhysicalDevice4, StartAddress=0mb, Size=17336mb),
(PhysicalDevice5, StartAddress=0mb, Size=17336mb),
(PhysicalDevice6, StartAddress=0mb, Size=17336mb),
(PhysicalDevice7, StartAddress=0mb, Size=17336mb),
(PhysicalDevice8, StartAddress=0mb, Size=17336mb),
(PhysicalDevice9, StartAddress=0mb, Size=17336mb),
(PhysicalDevice10, StartAddress=0mb, Size=17336mb),
(PhysicalDevice11, StartAddress=0mb, Size=17336mb),
(PhysicalDevice12, StartAddress=0mb, Size=17336mb),
(PhysicalDevice13, StartAddress=0mb, Size=17336mb);
IntermediateDevice1 = StripeSize=64kb, Raid=0, WriteThrough=1,
Size=242704mb,
(PhysicalDevice14, StartAddress=0mb, Size=17336mb),
(PhysicalDevice15, StartAddress=0mb, Size=17336mb),
(PhysicalDevice16, StartAddress=0mb, Size=17336mb),

```

(PhysicalDevice17, StartAddress=0mb, Size=17336mb),
(PhysicalDevice18, StartAddress=0mb, Size=17336mb),
(PhysicalDevice19, StartAddress=0mb, Size=17336mb),
(PhysicalDevice20, StartAddress=0mb, Size=17336mb),
(PhysicalDevice21, StartAddress=0mb, Size=17336mb),
(PhysicalDevice22, StartAddress=0mb, Size=17336mb),
(PhysicalDevice23, StartAddress=0mb, Size=17336mb),
(PhysicalDevice24, StartAddress=0mb, Size=17336mb),
(PhysicalDevice25, StartAddress=0mb, Size=17336mb),
(PhysicalDevice26, StartAddress=0mb, Size=17336mb),
(PhysicalDevice27, StartAddress=0mb, Size=17336mb);

IntermediateDevice2 = StripeSize=64kb, Raid=0, WriteThrough=1,
Size=242704mb,

(PhysicalDevice28, StartAddress=0mb, Size=17336mb),
(PhysicalDevice29, StartAddress=0mb, Size=17336mb),
(PhysicalDevice30, StartAddress=0mb, Size=17336mb),
(PhysicalDevice31, StartAddress=0mb, Size=17336mb),
(PhysicalDevice32, StartAddress=0mb, Size=17336mb),
(PhysicalDevice33, StartAddress=0mb, Size=17336mb),
(PhysicalDevice34, StartAddress=0mb, Size=17336mb),
(PhysicalDevice35, StartAddress=0mb, Size=17336mb),
(PhysicalDevice36, StartAddress=0mb, Size=17336mb),
(PhysicalDevice37, StartAddress=0mb, Size=17336mb),
(PhysicalDevice38, StartAddress=0mb, Size=17336mb),
(PhysicalDevice39, StartAddress=0mb, Size=17336mb),
(PhysicalDevice40, StartAddress=0mb, Size=17336mb),
(PhysicalDevice41, StartAddress=0mb, Size=17336mb);

LogicalDevice0 = StripeSize=64kb, Raid=12, WriteThrough=1,
Size=728112mb, BIOSGeometry=8GB,

Size=242704mb),
(IntermediateDevice0, StartAddress=0mb,
Size=242704mb),
(IntermediateDevice1, StartAddress=0mb,
Size=242704mb),
(IntermediateDevice2, StartAddress=0mb,
Size=242704mb);

EndGroup
BeginControllerParameter

```

```

ControllerName = eXtremeRAID 2000;

ControllerType = 28;

FirmwareVersion = 7.01;

CacheLineSize = 8KB;

BackgroundTaskRate = 50;

InitiatorID = 7;

DiskStartupMode = AutoSpin;

DevicesPerSpin = 2;

InitialDelay = 6S;

SequentialDelay = 0S;

EnableDriveSizing = 0;

EnableClustering = 0;

EnableBGInit = 1;

EnableReadAhead = 0;

EnableBiosLoadDelay = 0;

EnableForcedUnitAccess = 0;

DisableBios = 1;

EnableCDROMBoot = 0;

EnableStorageWorks = 0;

EnableSAFTE = 0;

EnableSES = 0;

EnableARM = 1;

EnableOFM = 1;

OEMCode = 0;

StartupOption = 0;

```

EndControllerParameter

End

Mylex eXtremeRAID 2000 Adapter 3

Begin

BeginGroup

PhysicalDevice0 = Channel=0, Target=0, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice1 = Channel=0, Target=1, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice2 = Channel=0, Target=2, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice3 = Channel=0, Target=3, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice4 = Channel=0, Target=4, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice5 = Channel=0, Target=5, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice6 = Channel=0, Target=6, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice7 = Channel=0, Target=8, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice8 = Channel=0, Target=9, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice9 = Channel=0, Target=10, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice10 = Channel=0, Target=11, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice11 = Channel=0, Target=12, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice12 = Channel=0, Target=13, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice13 = Channel=0, Target=14, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice14 = Channel=1, Target=0, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice15 = Channel=1, Target=1, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice16 = Channel=1, Target=2, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice17 = Channel=1, Target=3, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice18 = Channel=1, Target=4, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice19 = Channel=1, Target=5, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice20 = Channel=1, Target=6, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice21 = Channel=1, Target=8, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice22 = Channel=1, Target=9, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice23 = Channel=1, Target=10, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice24 = Channel=1, Target=11, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice25 = Channel=1, Target=12, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice26 = Channel=1, Target=13, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice27 = Channel=1, Target=14, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice28 = Channel=2, Target=0, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice29 = Channel=2, Target=1, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice30 = Channel=2, Target=2, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice31 = Channel=2, Target=3, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice32 = Channel=2, Target=4, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice33 = Channel=2, Target=5, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice34 = Channel=2, Target=6, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice35 = Channel=2, Target=8, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice36 = Channel=2, Target=9, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice37 = Channel=2, Target=10, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice38 = Channel=2, Target=11, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice39 = Channel=2, Target=12, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice40 = Channel=2, Target=13, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
PhysicalDevice41 = Channel=2, Target=14, Size=17340mb,
State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
IntermediateDevice0 = StripeSize=64kb, Raid=0, WriteThrough=1,
Size=242704mb,
(PhysicalDevice0, StartAddress=0mb, Size=17336mb),
(PhysicalDevice1, StartAddress=0mb, Size=17336mb),
(PhysicalDevice2, StartAddress=0mb, Size=17336mb),
(PhysicalDevice3, StartAddress=0mb, Size=17336mb),
(PhysicalDevice4, StartAddress=0mb, Size=17336mb),
(PhysicalDevice5, StartAddress=0mb, Size=17336mb),
(PhysicalDevice6, StartAddress=0mb, Size=17336mb),
(PhysicalDevice7, StartAddress=0mb, Size=17336mb),
(PhysicalDevice8, StartAddress=0mb, Size=17336mb),
(PhysicalDevice9, StartAddress=0mb, Size=17336mb),

<pre> (PhysicalDevice10, StartAddress=0mb, Size=17336mb), (PhysicalDevice11, StartAddress=0mb, Size=17336mb), (PhysicalDevice12, StartAddress=0mb, Size=17336mb), (PhysicalDevice13, StartAddress=0mb, Size=17336mb); IntermediateDevice1 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=242704mb, (PhysicalDevice14, StartAddress=0mb, Size=17336mb), (PhysicalDevice15, StartAddress=0mb, Size=17336mb), (PhysicalDevice16, StartAddress=0mb, Size=17336mb), (PhysicalDevice17, StartAddress=0mb, Size=17336mb), (PhysicalDevice18, StartAddress=0mb, Size=17336mb), (PhysicalDevice19, StartAddress=0mb, Size=17336mb), (PhysicalDevice20, StartAddress=0mb, Size=17336mb), (PhysicalDevice21, StartAddress=0mb, Size=17336mb), (PhysicalDevice22, StartAddress=0mb, Size=17336mb), (PhysicalDevice23, StartAddress=0mb, Size=17336mb), (PhysicalDevice24, StartAddress=0mb, Size=17336mb), (PhysicalDevice25, StartAddress=0mb, Size=17336mb), (PhysicalDevice26, StartAddress=0mb, Size=17336mb), (PhysicalDevice27, StartAddress=0mb, Size=17336mb); IntermediateDevice2 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=242704mb, (PhysicalDevice28, StartAddress=0mb, Size=17336mb), (PhysicalDevice29, StartAddress=0mb, Size=17336mb), (PhysicalDevice30, StartAddress=0mb, Size=17336mb), (PhysicalDevice31, StartAddress=0mb, Size=17336mb), (PhysicalDevice32, StartAddress=0mb, Size=17336mb), (PhysicalDevice33, StartAddress=0mb, Size=17336mb), (PhysicalDevice34, StartAddress=0mb, Size=17336mb), (PhysicalDevice35, StartAddress=0mb, Size=17336mb), (PhysicalDevice36, StartAddress=0mb, Size=17336mb), (PhysicalDevice37, StartAddress=0mb, Size=17336mb), (PhysicalDevice38, StartAddress=0mb, Size=17336mb), (PhysicalDevice39, StartAddress=0mb, Size=17336mb), (PhysicalDevice40, StartAddress=0mb, Size=17336mb), </pre>	<pre> (PhysicalDevice41, StartAddress=0mb, Size=17336mb); LogicalDevice0 = StripeSize=64kb, Raid=12, WriteThrough=1, Size=728112mb, BIOSGeometry=8GB, (IntermediateDevice0, StartAddress=0mb, Size=242704mb), (IntermediateDevice1, StartAddress=0mb, Size=242704mb), (IntermediateDevice2, StartAddress=0mb, Size=242704mb); EndGroup BeginControllerParameter ControllerName = eXtremeRAID 2000; ControllerType = 28; FirmwareVersion = 7.01; CacheLineSize = 8KB; BackgroundTaskRate = 50; InitiatorID = 7; DiskStartupMode = AutoSpin; DevicesPerSpin = 2; InitialDelay = 6S; SequentialDelay = 0S; EnableDriveSizing = 0; EnableClustering = 0; EnableBGInit = 1; EnableReadAhead = 0; EnableBiosLoadDelay = 0; EnableForcedUnitAccess = 0; DisableBios = 1; EnableCDROMBoot = 0; EnableStorageWorks = 0; EnableSAFTE = 0; EnableSES = 0; EnableARM = 1; EnableOFM = 1; OEMCode = 0; StartupOption = 0; </pre>
--	--

EndControllerParameter

End

Mylex eXtremeRAID 2000 Adapter 4

Begin

BeginGroup

PhysicalDevice0 = Channel=0, Target=0, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice1 = Channel=0, Target=1, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice2 = Channel=0, Target=2, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice3 = Channel=0, Target=3, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice4 = Channel=0, Target=4, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice5 = Channel=0, Target=5, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice6 = Channel=0, Target=6, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice7 = Channel=0, Target=8, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice8 = Channel=0, Target=9, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice9 = Channel=0, Target=10, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice10 = Channel=0, Target=11, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice11 = Channel=0, Target=12, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice12 = Channel=0, Target=13, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice13 = Channel=0, Target=14, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice14 = Channel=2, Target=0, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice15 = Channel=1, Target=0, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice16 = Channel=1, Target=1, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice17 = Channel=1, Target=2, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice18 = Channel=1, Target=3, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice19 = Channel=1, Target=4, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice20 = Channel=1, Target=5, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice21 = Channel=1, Target=6, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice22 = Channel=1, Target=8, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice23 = Channel=1, Target=9, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice24 = Channel=1, Target=10, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice25 = Channel=1, Target=11, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice26 = Channel=1, Target=12, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice27 = Channel=1, Target=13, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice28 = Channel=1, Target=14, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice29 = Channel=3, Target=8, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

IntermediateDevice0 = StripeSize=64kb, Raid=0, WriteThrough=1,
Size=260040mb,

(PhysicalDevice0, StartAddress=0mb, Size=17336mb),
(PhysicalDevice1, StartAddress=0mb, Size=17336mb),
(PhysicalDevice2, StartAddress=0mb, Size=17336mb),
(PhysicalDevice3, StartAddress=0mb, Size=17336mb),
(PhysicalDevice4, StartAddress=0mb, Size=17336mb),

(PhysicalDevice5, StartAddress=0mb, Size=17336mb),
(PhysicalDevice6, StartAddress=0mb, Size=17336mb),
(PhysicalDevice7, StartAddress=0mb, Size=17336mb),
(PhysicalDevice8, StartAddress=0mb, Size=17336mb),
(PhysicalDevice9, StartAddress=0mb, Size=17336mb),
(PhysicalDevice10, StartAddress=0mb, Size=17336mb),
(PhysicalDevice11, StartAddress=0mb, Size=17336mb),
(PhysicalDevice12, StartAddress=0mb, Size=17336mb),
(PhysicalDevice13, StartAddress=0mb, Size=17336mb),
(PhysicalDevice14, StartAddress=0mb, Size=17336mb);

IntermediateDevice1 = StripeSize=64kb, Raid=0, WriteThrough=1,
Size=260040mb,

(PhysicalDevice15, StartAddress=0mb, Size=17336mb),
(PhysicalDevice16, StartAddress=0mb, Size=17336mb),
(PhysicalDevice17, StartAddress=0mb, Size=17336mb),
(PhysicalDevice18, StartAddress=0mb, Size=17336mb),
(PhysicalDevice19, StartAddress=0mb, Size=17336mb),
(PhysicalDevice20, StartAddress=0mb, Size=17336mb),
(PhysicalDevice21, StartAddress=0mb, Size=17336mb),
(PhysicalDevice22, StartAddress=0mb, Size=17336mb),
(PhysicalDevice23, StartAddress=0mb, Size=17336mb),
(PhysicalDevice24, StartAddress=0mb, Size=17336mb),
(PhysicalDevice25, StartAddress=0mb, Size=17336mb),
(PhysicalDevice26, StartAddress=0mb, Size=17336mb),
(PhysicalDevice27, StartAddress=0mb, Size=17336mb),
(PhysicalDevice28, StartAddress=0mb, Size=17336mb),
(PhysicalDevice29, StartAddress=0mb, Size=17336mb);

LogicalDevice0 = StripeSize=64kb, Raid=12, WriteThrough=1,
Size=520080mb, BIOSGeometry=8GB,

(IntermediateDevice0, StartAddress=0mb,
Size=260040mb),

(IntermediateDevice1, StartAddress=0mb,
Size=260040mb);

PhysicalDevice30 = Channel=2, Target=1, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice31 = Channel=3, Target=9, Size=17340mb, State=Online, TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;	(PhysicalDevice30, StartAddress=0mb, Size=17336mb), (PhysicalDevice31, StartAddress=0mb, Size=17336mb);
PhysicalDevice32 = Channel=2, Target=2, Size=17340mb, State=Online, TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;	IntermediateDevice3 = StripeSize=64kb, Raid=1, WriteThrough=1, Size=17336mb, (PhysicalDevice32, StartAddress=0mb, Size=17336mb), (PhysicalDevice33, StartAddress=0mb, Size=17336mb);
PhysicalDevice33 = Channel=3, Target=10, Size=17340mb, State=Online, TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;	IntermediateDevice4 = StripeSize=64kb, Raid=1, WriteThrough=1, Size=17336mb, (PhysicalDevice34, StartAddress=0mb, Size=17336mb), (PhysicalDevice35, StartAddress=0mb, Size=17336mb);
PhysicalDevice34 = Channel=2, Target=3, Size=17340mb, State=Online, TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;	IntermediateDevice5 = StripeSize=64kb, Raid=1, WriteThrough=1, Size=17336mb, (PhysicalDevice36, StartAddress=0mb, Size=17336mb), (PhysicalDevice37, StartAddress=0mb, Size=17336mb);
PhysicalDevice35 = Channel=3, Target=11, Size=17340mb, State=Online, TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;	IntermediateDevice6 = StripeSize=64kb, Raid=1, WriteThrough=1, Size=17336mb, (PhysicalDevice38, StartAddress=0mb, Size=17336mb), (PhysicalDevice39, StartAddress=0mb, Size=17336mb);
PhysicalDevice36 = Channel=2, Target=4, Size=17340mb, State=Online, TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;	IntermediateDevice7 = StripeSize=64kb, Raid=1, WriteThrough=1, Size=17336mb, (PhysicalDevice40, StartAddress=0mb, Size=17336mb), (PhysicalDevice41, StartAddress=0mb, Size=17336mb);
PhysicalDevice37 = Channel=3, Target=12, Size=17340mb, State=Online, TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;	LogicalDevice1 = StripeSize=64kb, Raid=12, WriteThrough=1, Size=104016mb, BIOSGeometry=8GB, (IntermediateDevice2, StartAddress=0mb, Size=17336mb), (IntermediateDevice3, StartAddress=0mb, Size=17336mb), (IntermediateDevice4, StartAddress=0mb, Size=17336mb), (IntermediateDevice5, StartAddress=0mb, Size=17336mb), (IntermediateDevice6, StartAddress=0mb, Size=17336mb), (IntermediateDevice7, StartAddress=0mb, Size=17336mb);
PhysicalDevice38 = Channel=2, Target=5, Size=17340mb, State=Online, TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;	EndGroup
PhysicalDevice39 = Channel=3, Target=13, Size=17340mb, State=Online, TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;	BeginControllerParameter
PhysicalDevice40 = Channel=2, Target=6, Size=17340mb, State=Online, TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;	ControllerName = eXtremeRAID 2000;
PhysicalDevice41 = Channel=3, Target=14, Size=17340mb, State=Online, TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;	ControllerType = 28;
IntermediateDevice2 = StripeSize=64kb, Raid=1, WriteThrough=1, Size=17336mb,	

```

FirmwareVersion = 7.01;

CacheLineSize = 8KB;

BackgroundTaskRate = 50;

InitiatorID = 7;

DiskStartupMode = AutoSpin;

DevicesPerSpin = 2;

InitialDelay = 6S;

SequentialDelay = 0S;

EnableDriveSizing = 0;

EnableClustering = 0;

EnableBGInit = 1;

EnableReadAhead = 0;

EnableBiosLoadDelay = 0;

EnableForcedUnitAccess = 0;

DisableBios = 1;

EnableCDROMBoot = 0;

EnableStorageWorks = 0;

EnableSAFTE = 0;

EnableSES = 0;

EnableARM = 1;

EnableOFM = 1;

OEMCode = 0;

StartupOption = 0;

```

EndControllerParameter

End

Client Configuration Parameters

Microsoft Windows 2000 Server System Information Report

System Information report written at: 07/11/2002 02:23:30 PM
[System Information]

[Following are sub-categories of this main category]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Server
Version	5.0.2195 Service Pack 2 Build 2195
OS Manufacturer	Microsoft Corporation

System Name	CLIENT10
System Manufacturer	IBM
System Model	IBM eserver xSeries 220 -[4444aaa]-
System Type	X86-based PC
Processor	x86 Family 6 Model 11 Stepping 1 GenuineIntel ~1394 Mhz
Processor	x86 Family 6 Model 11 Stepping 1 GenuineIntel ~1394 Mhz
BIOS Version	IBM BIOS Ver 1.0
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition1
Locale	United States
User Name	CLIENT10\Administrator
Time Zone	Eastern Daylight Time
Total Physical Memory	654,808 KB
Available Physical Memory	565,656 KB
Total Virtual Memory	2,865,604 KB
Available Virtual Memory	2,697,904 KB
Page File Space	2,210,796 KB
Page File	C:\pagefile.sys

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

Resource	Device
No conflicted/shared resources	

[DMA]

Channel	Device	Status
2	Standard floppy disk controller	OK
4	Direct memory access controller	OK

[Forced Hardware]

Device	PNP Device ID
No Forced Hardware	

[I/O]

Address Range	Device	Status
0x0000-0x303F	PCI bus	OK
0x0000-0x303F	Direct memory access controller	OK
0x03B0-0x03BB	S3 Graphics Inc. Savage4	OK
0x03C0-0x03DF	S3 Graphics Inc. Savage4	OK
0x3000-0x303F	IBM 10/100 Ethernet Server Adapter	OK
0x2000-0x2FFF	DEC 21152 PCI to PCI bridge	OK
0x2000-0x2FFF	Intel(R) PRO/100+ Dual Port Server Adapter	OK
0x2020-0x203F	Intel(R) PRO/100+ Dual Port Server Adapter #2	OK
0x0A79-0x0A79	ISAPNP Read Data Port	OK
0x0279-0x0279	ISAPNP Read Data Port	OK
0x02F4-0x02F7	ISAPNP Read Data Port	OK
0x002E-0x002F	Motherboard resources	OK
0x0438-0x0439	Motherboard resources	OK
0x0430-0x0437	Motherboard resources	OK
0x0430-0x0437	Not Available	OK
0x0060-0x0060	Standard 101/102-Key or Microsoft Natural PS/2	
Keyboard	OK	
0x0064-0x0064	Standard 101/102-Key or Microsoft Natural PS/2	
Keyboard	OK	
0x03F0-0x03F5	Standard floppy disk controller	OK
0x03F7-0x03F7	Standard floppy disk controller	OK
0x0378-0x037F	Printer Port (LPT1)	OK
0x03F8-0x03FF	Communications Port (COM1)	OK

0x02F8-0x02FF	Communications Port (COM2)	OK
0x0020-0x0021	Advanced programmable interrupt controller	OK
0x00A0-0x00A1	Advanced programmable interrupt controller	OK
0x0080-0x008F	Direct memory access controller	OK
0x00C0-0x00DF	Direct memory access controller	OK
0x0040-0x0043	System timer	OK
0x0070-0x0073	System CMOS/real time clock	OK
0x0061-0x0061	System speaker	OK
0x00F0-0x00FF	Numeric data processor	OK
0x0600-0x0600	Motherboard resources	OK
0x0374-0x0375	Motherboard resources	OK
0x0377-0x0377	Motherboard resources	OK
0x0F50-0x0F58	Motherboard resources	OK
0x0700-0x070F	Standard Dual Channel PCI IDE Controller	OK
0x01F0-0x01F7	Primary IDE Channel	OK
0x03F6-0x03F6	Primary IDE Channel	OK
0x0170-0x0177	Secondary IDE Channel	OK
0x0376-0x0376	Secondary IDE Channel	OK
0x3040-0xFFFF	PCI bus	OK
0x3100-0x31FF	Adaptec AIC-7892 - Ultra160 SCSI	OK
0x4000-0x4FFF	DEC 21152 PCI to PCI bridge	OK
0x4000-0x4FFF	Intel(R) PRO/100+ Dual Port Server Adapter #3	OK
0x4020-0x403F	Intel(R) PRO/100+ Dual Port Server Adapter #4	OK

[IRQs]

IRQ Number	Device
30	Microsoft ACPI-Compliant System
31	S3 Graphics Inc. Savage4
27	IBM 10/100 Ethernet Server Adapter
16	Intel(R) PRO/100+ Dual Port Server Adapter
17	Intel(R) PRO/100+ Dual Port Server Adapter #2
1	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
12	PS/2 Compatible Mouse
6	Standard floppy disk controller
4	Communications Port (COM1)
3	Communications Port (COM2)
8	System CMOS/real time clock
13	Numeric data processor
14	Primary IDE Channel
10	Standard OpenHCD USB Host Controller
28	Adaptec AIC-7892 - Ultra160 SCSI
20	Intel(R) PRO/100+ Dual Port Server Adapter #3
21	Intel(R) PRO/100+ Dual Port Server Adapter #4

[Memory]

Range	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	S3 Graphics Inc. Savage4	OK
0xF0000000-0xFBFFFFFF	PCI bus	OK
0xF0000000-0xFBFFFFFF	S3 Graphics Inc. Savage4	OK
0xFC000000-0xFFFFFFFF	PCI bus	OK
0xFEB80000-0xFEBFFFFFF	S3 Graphics Inc. Savage4	OK
0xFEB7F000-0xFEB7FFFF	IBM 10/100 Ethernet Server Adapter	OK
0xFEB40000-0xFEB5FFFF	IBM 10/100 Ethernet Server Adapter	OK
0xFD000000-0xFE1FFFFF	DEC 21152 PCI to PCI bridge	OK
0xFBF00000-0xFBFFFFFF	DEC 21152 PCI to PCI bridge	OK
0xFBF00000-0xFBFFFFFF	Intel(R) PRO/100+ Dual Port Server Adapter	OK
0xFE000000-0xFE0FFFFF	Intel(R) PRO/100+ Dual Port Server Adapter	OK
0xFBF01000-0xFBF01FFF	Intel(R) PRO/100+ Dual Port Server Adapter #2	OK

0xFE100000-0xFE1FFFFF	Intel(R) PRO/100+ Dual Port Server Adapter #2	OK
0xFEB7E000-0xFEB7EFFF	Standard OpenHCD USB Host Controller	OK
0x28000000-0xECFFFFFF	PCI bus	OK
0xED000000-0xEFFFFFFF	PCI bus	OK
0xEFFFF000-0xEFFFFFFF	Adaptec AIC-7892 - Ultra160 SCSI	OK
0xEE000000-0xEF1FFFFF	DEC 21152 PCI to PCI bridge	OK
0xECF00000-0xECFFFFFF	DEC 21152 PCI to PCI bridge	OK
0xECF00000-0xECFFFFFF	Intel(R) PRO/100+ Dual Port Server Adapter #3	OK
0xEF000000-0xEF0FFFFF	Intel(R) PRO/100+ Dual Port Server Adapter #3	OK
0xECF01000-0xECF01FFF	Intel(R) PRO/100+ Dual Port Server Adapter #4	OK
0xEF100000-0xEF1FFFFF	Intel(R) PRO/100+ Dual Port Server Adapter #4	OK

[Components]

[Following are sub-categories of this main category]

[Multimedia]

[Following are sub-categories of this main category]

[Audio Codecs]

Codec Version	Manufacturer	Description	Status	File
c:\winnt\system32\msg723.acm	Microsoft Corporation	723	OK	
C:\WINNT\System32\MSG723.ACM		109,328 bytes	4.4.3385	106.77 KB
c:\winnt\system32\msgsm32.acm	Microsoft Corporation	32	OK	
C:\WINNT\System32\MSGSM32.ACM		22,27 KB (22,800 bytes)	5.00.2134.1	
c:\winnt\system32\msg711.acm	Microsoft Corporation	711	OK	
C:\WINNT\System32\MSG711.ACM		10,27 KB (10,512 bytes)	5.00.2134.1	
c:\winnt\system32\lhacm.acm	Microsoft Corporation	lhacm	OK	
C:\WINNT\System32\LHACM.ACM		34,064 bytes	4.4.3385	33.27 KB
c:\winnt\system32\msadp32.acm	Microsoft Corporation	32	OK	
C:\WINNT\System32\MSADP32.ACM		14,77 KB (15,120 bytes)	5.00.2134.1	
c:\winnt\system32\imaadp32.acm	Microsoft Corporation	32	OK	
C:\WINNT\System32\IMAADP32.ACM		16,27 KB (16,656 bytes)	5.00.2134.1	
c:\winnt\system32\tsoft32.acm	DSP GROUP, INC.	32	OK	
C:\WINNT\System32\TSOFT32.ACM		9,27 KB (9,488 bytes)	1.01	
c:\winnt\system32\iac25_32.ax	Intel Corporation	25_32	Indeo® audio software	
C:\WINNT\System32\IAC25_32.AX		199,680 bytes	2.05.53	195.00 KB

[Video Codecs]

Codec Version	Manufacturer	Description	Status	File
c:\winnt\system32\msh261.drv	Microsoft Corporation	261	OK	
C:\WINNT\System32\MSH261.DRV		167,696 bytes	4.4.3385	163.77 KB
c:\winnt\system32\msh263.drv	Microsoft Corporation	263	OK	
C:\WINNT\System32\MSH263.DRV		258,320 bytes	4.4.3385	252.27 KB

```

c:\winnt\system32\msvide32.dll  Microsoft Corporation
OK      C:\WINNT\System32\MSVIDC32.DLL  5.00.2134.1
27.27 KB (27,920 bytes)      12/7/1999 7:00:00 AM
c:\winnt\system32\msrle32.dll  Microsoft Corporation
OK      C:\WINNT\System32\MSRLE32.DLL  5.00.2134.1
10.77 KB (11,024 bytes)      12/7/1999 7:00:00 AM
c:\winnt\system32\ir32_32.dll  Intel(R) Corporation      OK
C:\WINNT\System32\IR32_32.DLL  Not Available      194.50 KB
(199,168 bytes)      12/7/1999 7:00:00 AM
c:\winnt\system32\ir50_32.dll  Intel Corporation      Indeo® video 5.10
OK      C:\WINNT\System32\IR50_32.DLL  R.5.10.15.2.55
737.50 KB (755,200 bytes)      12/7/1999 7:00:00 AM
c:\winnt\system32\iccvid.dll    Radius Inc.      OK
C:\WINNT\System32\ICCVID.DLL    1.10.0.6      108.00 KB (110,592
bytes)      12/7/1999 7:00:00 AM

```

[CD-ROM]

Item	Value
Drive	D:
Description	CD-ROM Drive
Media Loaded	False
Media Type	CD-ROM
Name	LG CD-ROM CRD-8484B
Manufacturer	(Standard CD-ROM drives)
Status	OK
Transfer Rate	Not Available
SCSI Target ID	0
PNP Device ID	
IDE\CDROMLG_CD-ROM_CRD-8484B	_____ 1.05 _____ \5
&326853DD&0&0.0.0	

[Sound Device]

Item	Value
No sound devices	

[Display]

Item	Value
Name	S3 Graphics Inc. Savage4
PNP Device ID	
PCI\VEN_5333&DEV_8A22&SUBSYS_01C51014&REV_06\3&267A616A	
&0&08	
Adapter Type	S3 Savage4, S3 Graphics Incorporated compatible
Adapter Description	S3 Graphics Inc. Savage4
Adapter RAM	8.00 MB (8,388,608 bytes)
Installed Drivers	s3savg4.dll
Driver Version	5.12.01.8013-8.40.02a
INF File	oem4.inf (S3SAVAGE4 section)
Color Planes	1
Color Table Entries	4294967296
Resolution	800 x 600 x 60 hertz
Bits/Pixel	32

[Infrared]

Item	Value
No infrared devices	

[Input]

[Following are sub-categories of this main category]

[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&389C1010&0
NumberOfFunctionKeys	12

[Pointing Device]

Item	Value
Hardware Type	PS/2 Compatible Mouse
Number of Buttons	3
Status	OK
PNP Device ID	ACPI\PNP0F13\4&389C1010&0
Power Management Supported	False
Double Click Threshold	6
Handedness	Right Handed Operation

[Modem]

Item	Value
No modems	

[Network]

[Following are sub-categories of this main category]

[Adapter]

Item	Value
Name	[00000000] Intel(R) PRO/100+ Dual Port Server Adapter
Adapter Type	Ethernet 802.3
Product Name	Intel(R) PRO/100+ Dual Port Server Adapter
Installed	True
PNP Device ID	
PCI\VEN_8086&DEV_1229&SUBSYS_10F08086&REV_05\4&273796BB&	
0&2048	
Last Reset	7/11/2002 8:47:44 AM
Index	0
Service Name	E100B
IP Address	192.168.110.10
IP Subnet	255.255.255.0
Default IP Gateway	Not Available
DHCP Enabled	False
DHCP Server	Not Available
DHCP Lease Expires	Not Available
DHCP Lease Obtained	Not Available
MAC Address	00:03:47:22:5D:1E
Service Name	E100B
IRQ Number	16
I/O Port	0x2000-0x2FFF
Driver	c:\winnt\system32\drivers\e100bnt5.sys (119056, 5.40.11.0000)

Name	[00000001] Intel(R) PRO/100+ Dual Port Server Adapter
Adapter Type	Ethernet 802.3
Product Name	Intel(R) PRO/100+ Dual Port Server Adapter
Installed	True
PNP Device ID	
PCI\VEN_8086&DEV_1229&SUBSYS_10F08086&REV_05\4&273796BB&	
0&2848	
Last Reset	7/11/2002 8:47:44 AM
Index	1
Service Name	E100B
IP Address	192.168.120.10
IP Subnet	255.255.255.0

Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:03:47:22:5D:1F
 Service Name E100B
 IRQ Number 17
 I/O Port 0x2020-0x203F
 Driver c:\winnt\system32\drivers\e100bnt5.sys (119056, 5.40.11.0000)

Name [00000002] RAS Async Adapter
 Adapter Type Not Available
 Product Name RAS Async Adapter
 Installed True
 PNP Device ID Not Available
 Last Reset 7/11/2002 8:47:44 AM
 Index 2
 Service Name AsyncMac
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Service Name Not Available

Name [00000003] WAN Miniport (L2TP)
 Adapter Type Not Available
 Product Name WAN Miniport (L2TP)
 Installed True
 PNP Device ID ROOT\MS_L2TPMINIPORT\0000
 Last Reset 7/11/2002 8:47:44 AM
 Index 3
 Service Name Rasl2tp
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Service Name Rasl2tp
 Driver c:\winnt\system32\drivers\rasl2tp.sys (50800, 5.00.2179.1)

Name [00000004] WAN Miniport (PPTP)
 Adapter Type Wide Area Network (WAN)
 Product Name WAN Miniport (PPTP)
 Installed True
 PNP Device ID ROOT\MS_PPTPMINIPORT\0000
 Last Reset 7/11/2002 8:47:44 AM
 Index 4
 Service Name PptpMiniport
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 50:50:54:50:30:30
 Service Name PptpMiniport
 Driver c:\winnt\system32\drivers\rasppptp.sys (47856, 5.00.2160.1)

Name [00000005] Direct Parallel

Adapter Type Not Available
 Product Name Direct Parallel
 Installed True
 PNP Device ID ROOT\MS_PTMINIPORT\0000
 Last Reset 7/11/2002 8:47:44 AM
 Index 5
 Service Name Raspti
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Service Name Raspti
 Driver c:\winnt\system32\drivers\raspti.sys (16880, 5.00.2146.1)

Name [00000006] WAN Miniport (IP)
 Adapter Type Not Available
 Product Name WAN Miniport (IP)
 Installed True
 PNP Device ID ROOT\MS_NDISWANIP\0000
 Last Reset 7/11/2002 8:47:44 AM
 Index 6
 Service Name NdisWan
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Service Name NdisWan
 Driver c:\winnt\system32\drivers\ndiswan.sys (90096, 5.00.2195.2779)

Name [00000007] IBM 10/100 Ethernet Server Adapter
 Adapter Type Ethernet 802.3
 Product Name IBM 10/100 Ethernet Server Adapter
 Installed True
 PNP Device ID PCI\VEN_8086&DEV_1229&SUBSYS_01F11014&REV_0C\3&267A616A&0&10
 Last Reset 7/11/2002 8:47:44 AM
 Index 7
 Service Name E100B
 IP Address 192.168.132.251
 IP Subnet 255.255.255.0
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:02:55:AA:02:D7
 Service Name E100B
 IRQ Number 27
 I/O Port 0x3000-0x303F
 Driver c:\winnt\system32\drivers\e100bnt5.sys (119056, 5.40.11.0000)

Name [00000008] Intel(R) PRO/100+ Dual Port Server Adapter
 Adapter Type Ethernet 802.3
 Product Name Intel(R) PRO/100+ Dual Port Server Adapter
 Installed True
 PNP Device ID PCI\VEN_8086&DEV_1229&SUBSYS_10F08086&REV_05\4&15F5026D&0&2028
 Last Reset 7/11/2002 8:47:44 AM

Index 8
 Service Name E100B
 IP Address 192.168.130.10
 IP Subnet 255.255.255.0
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:90:27:E8:71:60
 Service Name E100B
 IRQ Number 20
 I/O Port 0x4000-0x4FFF
 Driver c:\winnt\system32\drivers\e100bnt5.sys (119056, 5.40.11.0000)

Name [00000009] Intel(R) PRO/100+ Dual Port Server Adapter
 Adapter Type Ethernet 802.3
 Product Name Intel(R) PRO/100+ Dual Port Server Adapter
 Installed True
 PNP Device ID
 PCI\VEN_8086&DEV_1229&SUBSYS_10F08086&REV_05\4&15F5026D&0&2828
 Last Reset 7/11/2002 8:47:44 AM
 Index 9
 Service Name E100B
 IP Address 192.168.140.10
 IP Subnet 255.255.255.0
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:90:27:E8:71:61
 Service Name E100B
 IRQ Number 21
 I/O Port 0x4020-0x403F
 Driver c:\winnt\system32\drivers\e100bnt5.sys (119056, 5.40.11.0000)

[Protocol]

Item Value
 Name MSAFD Tcpip [TCP/IP]
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 16 bytes
 MaximumMessageSize 0 bytes
 MessageOriented False
 MinimumAddressSize 16 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData True
 SupportsGracefulClosing True
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD Tcpip [UDP/IP]
 ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 16 bytes
 MaximumMessageSize 65467 bytes
 MessageOriented True
 MinimumAddressSize 16 bytes

PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting True

Name RSVP UDP Service Provider
 ConnectionlessService True
 GuaranteesDelivery False
 GuaranteesSequencing False
 MaximumAddressSize 16 bytes
 MaximumMessageSize 65467 bytes
 MessageOriented True
 MinimumAddressSize 16 bytes
 PseudoStreamOriented False
 SupportsBroadcasting True
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption True
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting True

Name RSVP TCP Service Provider
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 16 bytes
 MaximumMessageSize 0 bytes
 MessageOriented False
 MinimumAddressSize 16 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption True
 SupportsExpeditedData True
 SupportsGracefulClosing True
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{44CF5AB0-B478-4D98-AD86-5CD3B62648C2}]
 SEQPACKET 6
 ConnectionlessService False
 GuaranteesDelivery True
 GuaranteesSequencing True
 MaximumAddressSize 20 bytes
 MaximumMessageSize 64000 bytes
 MessageOriented True
 MinimumAddressSize 20 bytes
 PseudoStreamOriented False
 SupportsBroadcasting False
 SupportsConnectData False
 SupportsDisconnectData False
 SupportsEncryption False
 SupportsExpeditedData False
 SupportsGracefulClosing False
 SupportsGuaranteedBandwidth False
 SupportsMulticasting False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{44CF5AB0-B478-4D98-AD86-5CD3B62648C2}]
 DATAGRAM 6

ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize20 bytes	
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{8380E329-FA5D-4971-8863-45389D633A46}]
 SEQPACKET 5

ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize20 bytes	
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{8380E329-FA5D-4971-8863-45389D633A46}]
 DATAGRAM 5

ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize20 bytes	
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{6FCF19B9-D1BC-4314-A154-2E811A2CC678}]
 SEQPACKET 4

ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes

MessageOriented	True
MinimumAddressSize20 bytes	
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{6FCF19B9-D1BC-4314-A154-2E811A2CC678}]
 DATAGRAM 4

ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize20 bytes	
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{151BB450-3D28-4525-B2D4-2331C74DCD26}]
 SEQPACKET 0

ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize20 bytes	
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{151BB450-3D28-4525-B2D4-2331C74DCD26}]
 DATAGRAM 0

ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize20 bytes	
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False

SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name MSAFD NetBIOS
[\Device\NetBT_Tcpip_{FA92D5FD-6E01-4F18-A447-837FEDE46296}]

SEQPACKET 1

ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name MSAFD NetBIOS
[\Device\NetBT_Tcpip_{FA92D5FD-6E01-4F18-A447-837FEDE46296}]

DATAGRAM 1

ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name MSAFD NetBIOS
[\Device\NetBT_Tcpip_{8F354C3B-893D-43F7-B7B0-3BF884D63BF1}]

SEQPACKET 2

ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name MSAFD NetBIOS
[\Device\NetBT_Tcpip_{8F354C3B-893D-43F7-B7B0-3BF884D63BF1}]

DATAGRAM 2

ConnectionlessService	True
-----------------------	------

GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name MSAFD NetBIOS
[\Device\NetBT_Tcpip_{9788862E-F4FC-434F-AEBB-56AA0E507EF7}]

SEQPACKET 3

ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name MSAFD NetBIOS
[\Device\NetBT_Tcpip_{9788862E-F4FC-434F-AEBB-56AA0E507EF7}]

DATAGRAM 3

ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

[WinSock]

Item	Value
File	c:\winnt\system32\winsock.dll
Version	3.10
Size	2.80 KB (2,864 bytes)

File	c:\winnt\system32\wssock32.dll
Version	5.00.2195.2871
Size	21.27 KB (21,776 bytes)

[Ports]

[Following are sub-categories of this main category]

[Serial]

Item	Value
Name	COM1
Status	OK
PNP Device ID	ACPI\PNP0501\1
Maximum Input Buffer Size	0
Maximum Output Buffer Size	False
Settable Baud Rate	True
Settable Data Bits	True
Settable Flow Control	True
Settable Parity	True
Settable Parity Check	True
Settable Stop Bits	True
Settable RLSD	True
Supports RLSD	True
Supports 16 Bit Mode	False
Supports Special Characters	False
Baud Rate	9600
Bits/Byte	8
Stop Bits	1
Parity	None
Busy	0
Abort Read/Write on Error	0
Binary Mode Enabled	-1
Continue XMit on XOff	0
CTS Outflow Control	0
Discard NULL Bytes	0
DSR Outflow Control	0
DSR Sensitivity	0
DTR Flow Control Type	Enable
EOF Character	0
Error Replace Character	0
Error Replacement Enabled	0
Event Character	0
Parity Check Enabled	0
RTS Flow Control Type	Enable
XOff Character	19
XOffXMit Threshold	512
XOn Character	17
XOnXMit Threshold	2048
XOnXOff InFlow Control	0
XOnXOff OutFlow Control	0
IRQ Number	4
I/O Port	0x03F8-0x03FF
Driver	c:\winnt\system32\drivers\serial.sys (62416, 5.00.2195.2780)

Name	COM2
Status	OK
PNP Device ID	ACPI\PNP0501\2
Maximum Input Buffer Size	0
Maximum Output Buffer Size	False
Settable Baud Rate	True
Settable Data Bits	True
Settable Flow Control	True
Settable Parity	True
Settable Parity Check	True
Settable Stop Bits	True
Settable RLSD	True
Supports RLSD	True
Supports 16 Bit Mode	False
Supports Special Characters	False
Baud Rate	9600

Bits/Byte	8
Stop Bits	1
Parity	None
Busy	0
Abort Read/Write on Error	0
Binary Mode Enabled	-1
Continue XMit on XOff	0
CTS Outflow Control	0
Discard NULL Bytes	0
DSR Outflow Control	0
DSR Sensitivity	0
DTR Flow Control Type	Enable
EOF Character	0
Error Replace Character	0
Error Replacement Enabled	0
Event Character	0
Parity Check Enabled	0
RTS Flow Control Type	Enable
XOff Character	19
XOffXMit Threshold	512
XOn Character	17
XOnXMit Threshold	2048
XOnXOff InFlow Control	0
XOnXOff OutFlow Control	0
IRQ Number	3
I/O Port	0x02F8-0x02FF
Driver	c:\winnt\system32\drivers\serial.sys (62416, 5.00.2195.2780)

[Parallel]

Item	Value
Name	LPT1
PNP Device ID	ACPI\PNP0400\1

[Storage]

[Following are sub-categories of this main category]

[Drives]

Item	Value
Drive	A:
Description	3 1/2 Inch Floppy Drive
Drive	C:
Description	Local Fixed Disk
Compressed	False
File System	NTFS
Size	8.46 GB (9,088,901,120 bytes)
Free Space	5.60 GB (6,011,867,136 bytes)
Volume Name	
Volume Serial Number	1C67EB25
Partition	Disk #0, Partition #0
Partition Size	8.46 GB (9,088,902,144 bytes)
Starting Offset	32256 bytes
Drive Description	Disk drive
Drive Manufacturer	(Standard disk drives)
Drive Model	IBM-PSG ST39204LC !# SCSI Disk Device
Drive BytesPerSector	512
Drive MediaLoaded	True
Drive MediaType	Fixed hard disk media
Drive Partitions	1
Drive SCSI Bus	0
Drive SCSI Logical Unit	0
Drive SCSI Port	2
Drive SCSI Target Id	0

Drive SectorsPerTrack 63
 Drive Size 9097159680 bytes
 Drive TotalCylinders 1106
 Drive TotalSectors 17767890
 Drive TotalTracks 282030
 Drive TracksPerCylinder 255

Drive E:
 Description Network Connection
 Provider Name \\192.168.132.253\e\$

[SCSI]

Item Value
 Name Adaptec AIC-7892 - Ultra160 SCSI
 Caption Adaptec AIC-7892 - Ultra160 SCSI
 Driver adpu160m
 Status OK
 PNP Device ID
 PCI\VEN_9005&DEV_008F&SUBSYS_008F1014&REV_02\3&13C0B0C5&0&18
 Device ID
 PCI\VEN_9005&DEV_008F&SUBSYS_008F1014&REV_02\3&13C0B0C5&0&18
 Device Map Not Available
 Index Not Available
 Max Number Controlled Not Available
 IRQ Number 28
 I/O Port 0x3100-0x31FF
 Driver c:\winnt\system32\drivers\adpu160m.sys (88000, d4.10 (4.10.4000))

[Printing]

Name Port Name Server Name
 No printing information

[Problem Devices]

Device PNP Device ID Error Code
 Not Available ACPI\IBM37D1\4&389C1010&0 28

[USB]

Device PNP Device ID
 Standard OpenHCD USB Host Controller
 PCI\VEN_1166&DEV_0220&SUBSYS_02201166&REV_04\3&267A616A&0&7A
 USB Root Hub USB\ROOT_HUB\4&372644EA&0

[Software Environment]

[Following are sub-categories of this main category]

[Drivers]

Name	Description	File	Type	Started	Start Mode
State	Status	Error Control	Accept Pause		Accept Stop
abiosdsk	Abiosdsk	Not Available	Kernel Driver		False
Disabled	Stopped	OK	Ignore	False	False
abp480n5	abp480n5	Not Available	Kernel Driver		False
Disabled	Stopped	OK	Normal	False	False
acpi	Microsoft ACPI Driver		c:\winnt\system32\drivers\acpi.sys		
Kernel Driver	True	Boot	Running	OK	Normal
False	True				

acpiec	ACPIEC	c:\winnt\system32\drivers\acpiec.sys	Kernel
Driver	False	Disabled Stopped OK	Normal False
adpu160m	adpu160m	c:\winnt\system32\drivers\adpu160m.sys	Kernel
Driver	True	Boot Running OK	Normal False
True			
afd	AFD Networking Support Environment		
c:\winnt\system32\drivers\afd.sys	Kernel Driver	True	Auto
Running	OK	Normal False	True
aha154x	Aha154x	Not Available	Kernel Driver False
Disabled	Stopped	OK	Normal False False
aic116x	aic116x	Not Available	Kernel Driver False
Disabled	Stopped	OK	Normal False False
aic78u2	aic78u2	Not Available	Kernel Driver False
Disabled	Stopped	OK	Normal False False
aic78xx	aic78xx	Not Available	Kernel Driver False
Disabled	Stopped	OK	Normal False False
ami0nt	ami0nt	Not Available	Kernel Driver False
Disabled	Stopped	OK	Normal False False
amsint	amsint	Not Available	Kernel Driver False
Disabled	Stopped	OK	Normal False False
asc	asc	Not Available	Kernel Driver False
Disabled	Stopped	OK	Normal False False
asc3350p	asc3350p	Not Available	Kernel Driver False
Disabled	Stopped	OK	Normal False False
asc3550	asc3550	Not Available	Kernel Driver False
Disabled	Stopped	OK	Normal False False
asynmac	RAS Asynchronous Media Driver		
c:\winnt\system32\drivers\asynmac.sys	Kernel Driver		False
Manual	Stopped	OK	Normal False False
atapi	Standard IDE/ESDI Hard Disk Controller		
c:\winnt\system32\drivers\atapi.sys	Kernel Driver		True
Boot	Running	OK	Normal False True
atdisk	Atdisk	Not Available	Kernel Driver False
Disabled	Stopped	OK	Ignore False False
atmarpc	ATM ARP Client Protocol		
c:\winnt\system32\drivers\atmarpc.sys	Kernel Driver		False
Manual	Stopped	OK	Normal False False
audstub	Audio Stub Driver	c:\winnt\system32\drivers\audstub.sys	
Kernel Driver	True	Manual Running OK	Normal
False	True		
beep	Beep	c:\winnt\system32\drivers\beep.sys	Kernel
Driver	True	System Running OK	Normal False
True			
buslogic	BusLogic	Not Available	Kernel Driver False
Disabled	Stopped	OK	Normal False False
cd20xrnt	cd20xrnt	Not Available	Kernel Driver False
Disabled	Stopped	OK	Normal False False
cdaudio	Cdaudio	c:\winnt\system32\drivers\cdaudio.sys	Kernel
Driver	False	System Stopped OK	Ignore False
False			
cdfs	Cdfs	c:\winnt\system32\drivers\cdfs.sys	File System
Driver	True	Disabled Running OK	Normal False
True			
cdrom	CD-ROM Driver	c:\winnt\system32\drivers\cdrom.sys	
Kernel Driver	True	System Running OK	Normal
False	True		
changer	Changer	Not Available	Kernel Driver False
System	Stopped	OK	Ignore False False
cpqarray	Cpqarray	Not Available	Kernel Driver False
Disabled	Stopped	OK	Normal False False
cpqarry2	cpqarry2	Not Available	Kernel Driver False
Disabled	Stopped	OK	Normal False False
cpqfcalm	cpqfcalm	Not Available	Kernel Driver False
Disabled	Stopped	OK	Normal False False
cpqfws2e	cpqfws2e	Not Available	Kernel Driver False
Disabled	Stopped	OK	Normal False False

dac960nt	dac960nt	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
deckzpsx	deckzpsx	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
dfsdriver	DfsDriver	c:\winnt\system32\drivers\dfs.sys	File System Driver	
True	Boot	Running	OK	Normal
disk	Disk Driver	c:\winnt\system32\drivers\disk.sys		
Kernel Driver	True	Boot	Running	OK
False	True			Normal
diskperf	Diskperf	c:\winnt\system32\drivers\diskperf.sys	Kernel	
Driver	True	Boot	Running	OK
True			Normal	False
dmboot	dmboot	c:\winnt\system32\drivers\dmboot.sys	Kernel	
Driver	False	Disabled	Stopped	OK
False			Normal	False
dmio	Logical Disk Manager Driver			
c:\winnt\system32\drivers\dmio.sys			Kernel Driver	True
Boot	Running	OK	Normal	False
dmload	dmload	c:\winnt\system32\drivers\dmload.sys	Kernel	
Driver	True	Boot	Running	OK
True			Normal	False
e100b	Intel(R) PRO Adapter Driver			
c:\winnt\system32\drivers\le100bnt5.sys			Kernel Driver	True
Manual	Running	OK	Normal	False
efs	EFS	c:\winnt\system32\drivers\efs.sys	File System Driver	
True	Disabled	Running	OK	Normal
fastfat	Fastfat	c:\winnt\system32\drivers\fastfat.sys	File System	
Driver	True	Disabled	Running	OK
True			Normal	False
fd16_700	Fd16_700	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
fdc	Floppy Disk Controller Driver			
Kernel Driver	True	Manual	Running	OK
False	True			Normal
fips	Fips	c:\winnt\system32\drivers\fips.sys	Kernel	
Driver	True	Auto	Running	OK
True			Normal	False
fireport	fireport	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
flashpnt	flashpnt	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
flpydisk	Floppy Disk Driver	c:\winnt\system32\drivers\flpydisk.sys		
Kernel Driver	True	Manual	Running	OK
False	True			Normal
ftdisk	Volume Manager Driver			
c:\winnt\system32\drivers\ftdisk.sys			Kernel Driver	True
Boot	Running	OK	Normal	False
gpc	Generic Packet Classifier			
c:\winnt\system32\drivers\msgpc.sys			Kernel Driver	True
Manual	Running	OK	Normal	False
i8042prt	i8042 Keyboard and PS/2 Mouse	Port Driver		
c:\winnt\system32\drivers\i8042prt.sys			Kernel Driver	True
System	Running	OK	Normal	False
ini910u	ini910u	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
intelide	IntelIde	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
ipfilterdriver	IP Traffic Filter Driver			
c:\winnt\system32\drivers\ipfltdrv.sys			Kernel Driver	False
Manual	Stopped	OK	Normal	False
ipinip	IP in IP Tunnel Driver			
c:\winnt\system32\drivers\ipinip.sys				
Kernel Driver	False	Manual	Stopped	OK
False	False			Normal
ipnat	IP Network Address Translator			
c:\winnt\system32\drivers\ipnat.sys				
Kernel Driver	False	Manual	Stopped	OK
False	False			Normal

ipsec	IPSEC driver	c:\winnt\system32\drivers\ipsec.sys		
Kernel Driver	True	Manual	Running	OK
False	True			Normal
ipsraidn	ipsraidn	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
isapnp	PnP ISA/EISA Bus Driver			
c:\winnt\system32\drivers\isapnp.sys			Kernel Driver	True
Boot	Running	OK	Critical	False
kbdclass	Keyboard Class Driver			
c:\winnt\system32\drivers\kbdclass.sys			Kernel Driver	True
System	Running	OK	Normal	False
ksecdd	KSecDD	c:\winnt\system32\drivers\ksecdd.sys		
Driver	True	Boot	Running	OK
True			Normal	False
lbrtfdc	lbrtfdc	Not Available	Kernel Driver	False
System	Stopped	OK	Ignore	False
lp6nds35	lp6nds35	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
mnmdm	mnmdm	c:\winnt\system32\drivers\mnmdm.sys	Kernel	
Driver	True	System	Running	OK
True			Ignore	False
modem	Modem	c:\winnt\system32\drivers\modem.sys	Kernel	
Driver	False	Manual	Stopped	OK
False			Ignore	False
mouclass	Mouse Class Driver	c:\winnt\system32\drivers\mouclass.sys		
Kernel Driver	True	System	Running	OK
False	True			Normal
mountmgr	MountMgr	c:\winnt\system32\drivers\mountmgr.sys	Kernel	
Driver	True	Boot	Running	OK
True			Normal	False
mraid35x	mraid35x	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
mrxsmb	MRXSMB	c:\winnt\system32\drivers\mrxsmb.sys	File System	
Driver	True	System	Running	OK
True			Normal	False
msfs	Msfs	c:\winnt\system32\drivers\msfs.sys	File System	
Driver	True	System	Running	OK
True			Normal	False
mskssrv	Microsoft Streaming Service Proxy			
c:\winnt\system32\drivers\mskssrv.sys			Kernel Driver	False
Manual	Stopped	OK	Normal	False
mspclock	Microsoft Streaming Clock Proxy			
c:\winnt\system32\drivers\mspclock.sys			Kernel Driver	False
Manual	Stopped	OK	Normal	False
mspqm	Microsoft Streaming Quality Manager Proxy			
c:\winnt\system32\drivers\mspqm.sys			Kernel Driver	False
Manual	Stopped	OK	Normal	False
mup	Mup	c:\winnt\system32\drivers\mup.sys	File System	
Driver	True	Boot	Running	OK
True			Normal	False
nrc710	Nrc710	Not Available	Kernel Driver	False
Disabled	Stopped	OK	Normal	False
ndis	NDIS System Driver			
c:\winnt\system32\drivers\ndis.sys				
Kernel Driver	True	Boot	Running	OK
False	True			Normal
ndistapi	Remote Access NDIS TAPI Driver			
c:\winnt\system32\drivers\ndistapi.sys			Kernel Driver	True
Manual	Running	OK	Normal	False
ndiswan	Remote Access NDIS WAN Driver			
c:\winnt\system32\drivers\ndiswan.sys			Kernel Driver	True
Manual	Running	OK	Normal	False
ndproxy	NDIS Proxy	c:\winnt\system32\drivers\ndproxy.sys		
Kernel Driver	True	Manual	Running	OK
False	True			Normal
netbios	NetBIOS Interface			
c:\winnt\system32\drivers\netbios.sys				
File System Driver	True	System	Running	OK
False	True			Normal

netbt	NetBios over Tcpip	c:\winnt\system32\drivers\netbt.sys	Kernel Driver	True	System	Running	OK	Normal	False
netdetect	NetDetect	c:\winnt\system32\drivers\netdetect.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	False
npfs	Npfs	c:\winnt\system32\drivers\npfs.sys	File System	True	System	Running	OK	Normal	False
ntfs	Ntfs	c:\winnt\system32\drivers\ntfs.sys	File System	True	Disabled	Running	OK	Normal	False
null	Null	c:\winnt\system32\drivers\null.sys	Kernel Driver	True	System	Running	OK	Normal	False
nwlkflt	IPX Traffic Filter Driver	c:\winnt\system32\drivers\nwlkflt.sys	Kernel Driver	Manual	Stopped	OK	Normal	False	False
nwlkfw	IPX Traffic Forwarder Driver	c:\winnt\system32\drivers\nwlkfw.sys	Kernel Driver	Manual	Stopped	OK	Normal	False	False
openhci	Microsoft USB Open Host Controller Driver	c:\winnt\system32\drivers\openhci.sys	Kernel Driver	Manual	Running	OK	Normal	False	True
parallel	Parallel class driver	c:\winnt\system32\drivers\parallel.sys	Kernel Driver	False	True	Manual	Running	OK	Normal
parport	Parallel port driver	c:\winnt\system32\drivers\parport.sys	Kernel Driver	False	True	System	Running	OK	Ignore
partmgr	PartMgr	c:\winnt\system32\drivers\partmgr.sys	Kernel Driver	True	Boot	Running	OK	Normal	False
parvdm	ParVdm	c:\winnt\system32\drivers\parvdm.sys	Kernel Driver	True	Auto	Running	OK	Ignore	False
pci	PCI Bus Driver	c:\winnt\system32\drivers\pci.sys	Kernel Driver	True	Boot	Running	OK	Critical	False
pcidump	PCIDump	Not Available	Kernel Driver	System	Stopped	OK	Ignore	False	False
pciide	PCIIde	c:\winnt\system32\drivers\pciide.sys	Kernel Driver	True	Boot	Running	OK	Normal	False
pcmcia	Pcmcia	c:\winnt\system32\drivers\pcmcia.sys	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
pdcomp	PDCOMP	Not Available	Kernel Driver	Manual	Stopped	OK	Ignore	False	False
pdframe	PDFRAME	Not Available	Kernel Driver	False	Manual	Stopped	OK	Ignore	False
pdreli	PDRELI	Not Available	Kernel Driver	Manual	Stopped	OK	Ignore	False	False
pdrframe	PDRFRAME	Not Available	Kernel Driver	False	Manual	Stopped	OK	Ignore	False
pptpminiport	WAN Miniport (PPTP)	c:\winnt\system32\drivers\rasppptp.sys	Kernel Driver	Manual	Running	OK	Normal	False	True
ptilink	Direct Parallel Link Driver	c:\winnt\system32\drivers\ptilink.sys	Kernel Driver	Manual	Running	OK	Normal	False	True
ql1080	ql1080	Not Available	Kernel Driver	Disabled	Stopped	OK	Normal	False	False
ql10wnt	ql10wnt	Not Available	Kernel Driver	Disabled	Stopped	OK	Normal	False	False
ql1240	ql1240	Not Available	Kernel Driver	Disabled	Stopped	OK	Normal	False	False
ql2100	ql2100	Not Available	Kernel Driver	Disabled	Stopped	OK	Normal	False	False
rasacd	Remote Access Auto Connection Driver	c:\winnt\system32\drivers\rasacd.sys	Kernel Driver	System	Running	OK	Normal	False	True
rasl2tp	WAN Miniport (L2TP)	c:\winnt\system32\drivers\rasl2tp.sys	Kernel Driver	Manual	Running	OK	Normal	False	True
raspti	Direct Parallel	c:\winnt\system32\drivers\raspti.sys	Kernel Driver	False	True	Manual	Running	OK	Normal
rca	Microsoft Streaming Network Raw Channel Access	c:\winnt\system32\drivers\rca.sys	Kernel Driver	Stopped	OK	Normal	False	False	Manual
rdbs	Rdbss	c:\winnt\system32\drivers\rdbs.sys	File System	Driver	True	System	Running	OK	Normal
rdpwd	RDPWD	c:\winnt\system32\drivers\rdpwd.sys	Kernel Driver	Driver	False	Manual	Stopped	OK	Ignore
redbook	Digital CD Audio Playback Filter Driver	c:\winnt\system32\drivers\redbook.sys	Kernel Driver	System	Stopped	OK	Normal	False	False
s3inc	S3Inc	c:\winnt\system32\drivers\s3sav4m.sys	Kernel Driver	Driver	False	Manual	Stopped	OK	Ignore
s3savage4	S3SAVAGE4	c:\winnt\system32\drivers\s3sav4m.sys	Kernel Driver	False	True	Manual	Running	OK	Ignore
serenum	Serenum Filter Driver	c:\winnt\system32\drivers\serenum.sys	Kernel Driver	False	True	Manual	Running	OK	Normal
serial	Serial port driver	c:\winnt\system32\drivers\serial.sys	Kernel Driver	False	True	System	Running	OK	Ignore
sfloppy	Sfloppy	c:\winnt\system32\drivers\sfloppy.sys	Kernel Driver	Driver	False	System	Stopped	OK	Ignore
sglfb	sglfb	Not Available	Kernel Driver	System	Stopped	OK	Normal	False	False
simbad	Simbad	Not Available	Kernel Driver	Disabled	Stopped	OK	Normal	False	False
sparrow	Sparrow	Not Available	Kernel Driver	Disabled	Stopped	OK	Normal	False	False
spud	Special Purpose Utility Driver	c:\winnt\system32\drivers\spud.sys	Kernel Driver	False	True	Manual	Running	OK	Normal
srv	Srv	c:\winnt\system32\drivers\srv.sys	File System Driver	True	Manual	Running	OK	Normal	True
swenum	Software Bus Driver	c:\winnt\system32\drivers\swenum.sys	Kernel Driver	False	True	Manual	Running	OK	Normal
symc810	symc810	Not Available	Kernel Driver	Disabled	Stopped	OK	Normal	False	False
symc8xx	symc8xx	Not Available	Kernel Driver	Disabled	Stopped	OK	Normal	False	False
sym_hi	sym_hi	Not Available	Kernel Driver	Disabled	Stopped	OK	Normal	False	False
tcpip	TCP/IP Protocol Driver	c:\winnt\system32\drivers\tcpip.sys	Kernel Driver	False	True	System	Running	OK	Normal
tdasync	TDASYN	c:\winnt\system32\drivers\tdasync.sys	Kernel Driver	False	False	Manual	Stopped	OK	Ignore
tdipx	TDIPX	c:\winnt\system32\drivers\tdipx.sys	Kernel Driver	Driver	False	Manual	Stopped	OK	Ignore

tdnetb	TDNETB	c:\winnt\system32\drivers\tdnetb.sys	Kernel
Driver	False	Manual Stopped OK Ignore	False
False			
tdpipe	TDPIPE	c:\winnt\system32\drivers\tdpipe.sys	Kernel
Driver	False	Manual Stopped OK Ignore	False
False			
tdspix	TDSPX	c:\winnt\system32\drivers\tdspix.sys	Kernel
Driver	False	Manual Stopped OK Ignore	False
False			
tdtcp	TDTCP	c:\winnt\system32\drivers\tdtcp.sys	Kernel
Driver	False	Manual Stopped OK Ignore	False
False			
termdd	Terminal Device Driver		
	c:\winnt\system32\drivers\termdd.sys	Kernel Driver	False
Disabled	Stopped	OK Normal False False	
tga	tga	Not Available	Kernel Driver
System	Stopped	OK Ignore False False	False
udfs	Udfs	c:\winnt\system32\drivers\udfs.sys	File System
Driver	False	Disabled Stopped OK Normal	False
False			
ultra66	ultra66	Not Available	Kernel Driver
Disabled	Stopped	OK Normal False False	False
update	Microcode Update Driver		
	c:\winnt\system32\drivers\update.sys	Kernel Driver	True
Manual	Running	OK Normal False True	
usbhub	Microsoft USB Standard Hub Driver		
	c:\winnt\system32\drivers\usbhub.sys	Kernel Driver	True
Manual	Running	OK Normal False True	
vgasave	VgaSave	c:\winnt\system32\drivers\vga.sys	Kernel
Driver	True	System Running OK Ignore	False
True			
wanarp	Remote Access IP ARP Driver		
	c:\winnt\system32\drivers\wanarp.sys	Kernel Driver	True
Manual	Running	OK Normal False True	
wdica	WDICA	Not Available	Kernel Driver
Manual	Stopped	OK Ignore False False	False
nmfcfg	NIC Management Service Configuration Driver		
	\\?\c:\winnt\system32\drivers\nmfcfg.sys	Kernel Driver	True
Manual	Running	OK Normal False True	

[Environment Variables]

Variable	Value	User Name
ComSpec	%SystemRoot%\system32\cmd.exe	<SYSTEM>
Os2LibPath	%SystemRoot%\system32\os2\dll;	<SYSTEM>
Path	%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;	
	C:\Program Files\Microsoft SQL Server\80\Tools\BINN	<SYSTEM>
windir	%SystemRoot%	<SYSTEM>
OS	Windows_NT	<SYSTEM>
PROCESSOR_ARCHITECTURE	x86	<SYSTEM>
PROCESSOR_LEVEL	6	<SYSTEM>
PROCESSOR_IDENTIFIER	x86 Family 6 Model 11 Stepping 1,	
GenuineIntel	<SYSTEM>	
PROCESSOR_REVISION	0b01	<SYSTEM>
NUMBER_OF_PROCESSORS	2	<SYSTEM>
PATHEXT		
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH		<SYSTEM>
TEMP	%SystemRoot%\TEMP	<SYSTEM>
TMP	%SystemRoot%\TEMP	<SYSTEM>
TEMP	%USERPROFILE%\Local Settings\Temp	
CLIENT10\Administrator		
TMP	%USERPROFILE%\Local Settings\Temp	
CLIENT10\Administrator		

[Jobs]

[Following are sub-categories of this main category]

[Print]

Document	Size	Owner	Notify	Status	Time Submitted
Start Time	Until Time	Elapsed Time	Pages Printed	Job ID	
Priority	Parameters	Driver Name	Print Processor	Host Print	
Queue	Data Type	Name			

No print jobs

[Network Connections]

Local Name	Remote Name	Type	Status	User Name
E:	\\192.168.132.253\e\$	Disk	OK	
CLIENT10\Administrator				

[Running Tasks]

Name	Path	Process ID	Priority	Min Working Set	Max
Working Set	Start Time	Version	Size	File Date	
system idle process	Not Available	0	0	Not	
Available	Not Available	Not Available	Unknown	Unknown	
Unknown					
system	Not Available	8	8	0	1413120
Not Available	Unknown	Unknown	Unknown		
smss.exe	c:\winnt\system32\smss.exe	164	11	204800	
1413120	7/11/2002 12:47:59 PM	5.00.2195.2901	44.27 KB		
(45,328 bytes)	12/7/1999 7:00:00 AM				
csrss.exe	Not Available	188	13	Not Available	
Not Available	7/11/2002 12:48:03 PM	Unknown	Unknown		
Unknown					
winlogon.exe	c:\winnt\system32\winlogon.exe	184	13		
204800	1413120 7/11/2002 12:48:05 PM	5.00.2195.2953			
173.77 KB (177,936 bytes)	12/7/1999 7:00:00 AM				
services.exe	c:\winnt\system32\services.exe	236	9		
204800	1413120 7/11/2002 12:48:06 PM	5.00.2195.2780			
86.77 KB (88,848 bytes)	12/7/1999 7:00:00 AM				
lsass.exe	c:\winnt\system32\lsass.exe	248	9	204800	
1413120	7/11/2002 12:48:06 PM	5.00.2195.2964	32.77 KB		
(33,552 bytes)	12/7/1999 7:00:00 AM				
svchost.exe	c:\winnt\system32\svchost.exe	416	8		
204800	1413120 7/11/2002 12:48:09 PM	5.00.2134.1			
7.77 KB (7,952 bytes)	12/7/1999 7:00:00 AM				
spoolsv.exe	c:\winnt\system32\spoolsv.exe	448	8		
204800	1413120 7/11/2002 12:48:10 PM	5.00.2161.1			
43.77 KB (44,816 bytes)	9/17/2001 9:39:14 AM				
msdtc.exe	c:\winnt\system32\msdtc.exe	476	8	204800	
1413120	7/11/2002 12:48:10 PM	1999.9.3421.3	6.77 KB		
(6,928 bytes)	9/17/2001 10:04:52 AM				
svchost.exe	c:\winnt\system32\svchost.exe	608	8		
204800	1413120 7/11/2002 12:48:12 PM	5.00.2134.1			
7.77 KB (7,952 bytes)	12/7/1999 7:00:00 AM				
llssrv.exe	c:\winnt\system32\llssrv.exe	636	9	204800	
1413120	7/11/2002 12:48:13 PM	5.00.2195.2649	114.27 KB		
(117,008 bytes)	5/4/2001 12:05:02 PM				
nmssvc.exe	c:\winnt\system32\nmssvc.exe	668	8		
204800	1413120 7/11/2002 12:48:13 PM	1.64.0.0	1012.00 KB		
(1,036,288 bytes)	10/8/2001 4:31:07 PM				
regsvc.exe	c:\winnt\system32\regsvc.exe	828	8	204800	
1413120	7/11/2002 12:48:26 PM	5.00.2195.2104	65.27 KB		
(66,832 bytes)	9/24/2001 10:26:33 AM				
mstask.exe	c:\winnt\system32\mstask.exe	852	8	204800	
1413120	7/11/2002 12:48:26 PM	4.71.2195.1	115.27 KB		
(118,032 bytes)	9/24/2001 10:26:26 AM				
tcpvcs.exe	c:\winnt\system32\tcpvcs.exe	888	8		
204800	1413120 7/11/2002 12:48:26 PM	5.00.2134.1			
24.77 KB (25,360 bytes)	12/7/1999 7:00:00 AM				

winmgmt.exe	c:\winnt\system32\wbem\winmgmt.exe	924	8	msinfo32.dll	5.00.2177.1	312.27 KB (319,760 bytes)
204800 1413120	7/11/2002 12:48:27 PM	1.50.1085.0029		9/17/2001 2:09:30 PM	Microsoft Corporation	c:\program
192.08 KB (196,685 bytes)	9/24/2001 10:26:43 AM			files\common files\microsoft shared\msinfo\msinfo32.dll		
inetinfo.exe	c:\winnt\system32\inetrv\inetinfo.exe	956	8	riched20.dll	5.30.23.1205	421.27 KB (431,376 bytes)
204800 1413120	7/11/2002 12:48:27 PM	5.00.0984	14.27 KB	9/24/2001 10:26:33 AM	Microsoft Corporation	
(14,608 bytes)	9/24/2001 10:27:34 AM			c:\winnt\system32\riched20.dll		
dfssvc.exe	c:\winnt\system32\dfssvc.exe	792	8	riched32.dll	5.00.2134.1	3.77 KB (3,856 bytes)
1413120 7/11/2002 12:48:31 PM	5.00.2195.2841	204800	88.27 KB	7:00:00 AM	Microsoft Corporation	12/7/1999
(90,384 bytes)	9/24/2001 10:26:13 AM			c:\winnt\system32\riched32.dll		
svchost.exe	c:\winnt\system32\svchost.exe	1156	8	els.dll	5.00.2175.1	151.27 KB (154,896 bytes)
204800 1413120	7/11/2002 12:49:10 PM	5.00.2134.1		7:00:00 AM	Microsoft Corporation	12/7/1999
7.77 KB (7,952 bytes)	12/7/1999 7:00:00 AM			c:\winnt\system32\els.dll		
explorer.exe	c:\winnt\explorer.exe	844	8	ntsmmgr.dll	1,0,0,1	427.77 KB (438,032 bytes)
1413120 7/11/2002 1:00:20 PM	5.00.3315.2846	204800	237.27 KB	7:00:00 AM	Microsoft Corporation and HighGround Systems, Inc.	12/7/1999
(242,960 bytes)	9/24/2001 10:26:39 AM			c:\winnt\system32\ntsmmgr.dll		
promon.exe	c:\winnt\system32\promon.exe	1252	8	mmfutil.dll	5.0.1085.0000	32.06 KB (32,829 bytes)
204800 1413120	7/11/2002 1:00:21 PM	4.08	30.50 KB	7:00:00 AM	Microsoft Corporation	12/7/1999
(31,232 bytes)	10/8/2001 4:31:08 PM			c:\winnt\system32\mmfutil.dll		
mmc.exe	c:\winnt\system32\mmc.exe	972	8	logdrive.dll	1.50.1085.0000	200.06 KB (204,863 bytes)
1413120 7/11/2002 2:19:03 PM	5.00.2195.2301	204800	589.27 KB	12/7/1999 7:00:00 AM	Microsoft Corporation	
(603,408 bytes)	9/24/2001 10:26:20 AM			c:\winnt\system32\logdrive.dll		
rsvp.exe	c:\winnt\system32\rsvp.exe	1344	8	dfrgres.dll	5.00.2150.1	27.50 KB (28,160 bytes)
1413120 7/11/2002 2:21:23 PM	5.00.2167.1	204800	172.77 KB	7:00:00 AM	Executive Software International, Inc.	12/7/1999
(176,912 bytes)	12/7/1999 7:00:00 AM			c:\winnt\system32\dfrgres.dll		

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
traffic.dll	5.00.2139.1	30.77 KB (31,504 bytes)	12/7/1999		
7:00:00 AM	Microsoft Corporation				
c:\winnt\system32\traffic.dll					
rsvp.exe	5.00.2167.1	172.77 KB (176,912 bytes)	12/7/1999		
7:00:00 AM	Microsoft Corporation				
c:\winnt\system32\rsvp.exe					
wbemprox.dll	1.50.1085.0045	40.08 KB (41,040 bytes)			
9/24/2001 10:26:43 AM	Microsoft Corporation				
c:\winnt\system32\wbem\wbemprox.dll					
mlang.dll	5.00.3103.1000	510.77 KB (523,024 bytes)	9/24/2001		
10:26:20 AM	Microsoft Corporation				
c:\winnt\system32\mlang.dll					
rassapi.dll	5.00.2188.1	14.27 KB (14,608 bytes)	12/7/1999		
7:00:00 AM	Microsoft Corporation				
c:\winnt\system32\rassapi.dll					
adsnt.dll	5.00.2195.2778	195.27 KB (199,952 bytes)	9/24/2001		
10:26:09 AM	Microsoft Corporation				
c:\winnt\system32\adsnt.dll					
dbghelp.dll	5.00.2195.2104	159.27 KB (163,088 bytes)			
5/4/2001 12:05:02 PM	Microsoft Corporation				
c:\winnt\system32\dbghelp.dll					
localsec.dll	5.00.2195.2130	230.27 KB (235,792 bytes)			
9/24/2001 10:26:19 AM	Microsoft Corporation				
c:\winnt\system32\localsec.dll					
devmgr.dll	5.00.2166.1	215.77 KB (220,944 bytes)	12/7/1999		
7:00:00 AM	Microsoft Corporation				
c:\winnt\system32\devmgr.dll					
filemgmt.dll	5.00.2195.2165	287.27 KB (294,160 bytes)			
9/24/2001 10:26:16 AM	Microsoft Corporation				
c:\winnt\system32\filemgmt.dll					
pdh.dll	5.00.2195.2739	147.77 KB (151,312 bytes)	9/24/2001		
10:26:32 AM	Microsoft Corporation				
c:\winnt\system32\pdh.dll					
smlogcfg.dll	5.00.2195.2485	273.27 KB (279,824 bytes)			
9/24/2001 10:26:35 AM	Microsoft Corporation				
c:\winnt\system32\smlogcfg.dll					
cabinet.dll	5.00.2147.1	54.77 KB (56,080 bytes)	12/7/1999		
7:00:00 AM	Microsoft Corporation				
c:\winnt\system32\cabinet.dll					

msinfo32.dll	5.00.2177.1	312.27 KB (319,760 bytes)
9/17/2001 2:09:30 PM	Microsoft Corporation	c:\program
files\common files\microsoft shared\msinfo\msinfo32.dll		
riched20.dll	5.30.23.1205	421.27 KB (431,376 bytes)
9/24/2001 10:26:33 AM	Microsoft Corporation	
c:\winnt\system32\riched20.dll		
riched32.dll	5.00.2134.1	3.77 KB (3,856 bytes)
7:00:00 AM	Microsoft Corporation	12/7/1999
c:\winnt\system32\riched32.dll		
els.dll	5.00.2175.1	151.27 KB (154,896 bytes)
7:00:00 AM	Microsoft Corporation	12/7/1999
c:\winnt\system32\els.dll		
ntsmmgr.dll	1,0,0,1	427.77 KB (438,032 bytes)
7:00:00 AM	Microsoft Corporation and HighGround Systems, Inc.	12/7/1999
c:\winnt\system32\ntsmmgr.dll		
mmfutil.dll	5.0.1085.0000	32.06 KB (32,829 bytes)
7:00:00 AM	Microsoft Corporation	12/7/1999
c:\winnt\system32\mmfutil.dll		
logdrive.dll	1.50.1085.0000	200.06 KB (204,863 bytes)
12/7/1999 7:00:00 AM	Microsoft Corporation	
c:\winnt\system32\logdrive.dll		
dfrgres.dll	5.00.2150.1	27.50 KB (28,160 bytes)
7:00:00 AM	Executive Software International, Inc.	12/7/1999
c:\winnt\system32\dfrgres.dll		
dfrgsnap.dll	5.00.2195.2104	41.77 KB (42,768 bytes)
9/24/2001 10:26:13 AM	Executive Software International, Inc.	
c:\winnt\system32\dfrgsnap.dll		
dmdskres.dll	2195.2104.297.3	119.50 KB (122,368 bytes)
9/24/2001 10:26:14 AM	Microsoft Corp., VERITAS Software	
c:\winnt\system32\dmdskres.dll		
dmutil.dll	2195.2104.297.3	42.27 KB (43,280 bytes)
10:26:14 AM	VERITAS Software Corp.	9/24/2001
c:\winnt\system32\dmutil.dll		
ntmsapi.dll	5.00.1948.1	51.77 KB (53,008 bytes)
10:26:29 AM	Microsoft Corporation	9/24/2001
c:\winnt\system32\ntmsapi.dll		
dmdskmgr.dll	2215.2215.297.3	160.27 KB (164,112 bytes)
9/24/2001 10:26:13 AM	Microsoft Corp., VERITAS Software	
c:\winnt\system32\dmdskmgr.dll		
mycomput.dll	5.00.2134.1	107.77 KB (110,352 bytes)
12/7/1999 7:00:00 AM	Microsoft Corporation	
c:\winnt\system32\mycomput.dll		
mmcmdmgr.dll	5.00.2178.1	815.27 KB (834,832 bytes)
12/7/1999 7:00:00 AM	Microsoft Corporation	
c:\winnt\system32\mmcmdmgr.dll		
mmc.exe	5.00.2195.2301	589.27 KB (603,408 bytes)
10:26:20 AM	Microsoft Corporation	9/24/2001
c:\winnt\system32\mmc.exe		
promon.exe	4.08	30.50 KB (31,232 bytes)
4:31:08 PM	Intel Corporation	10/8/2001
c:\winnt\system32\promon.exe		
wininet.dll	5.00.3315.1000	456.77 KB (467,728 bytes)
10:26:38 AM	Microsoft Corporation	9/24/2001
c:\winnt\system32\wininet.dll		
netplwiz.dll	5.00.2195.2370	169.77 KB (173,840 bytes)
9/24/2001 10:26:28 AM	Microsoft Corporation	
c:\winnt\system32\netplwiz.dll		
netmsg.dll	5.00.2137.1	152.50 KB (156,160 bytes)
7:00:00 AM	Microsoft Corporation	12/7/1999
c:\winnt\system32\netmsg.dll		
netui2.dll	5.00.2134.1	280.27 KB (286,992 bytes)
7:00:00 AM	Microsoft Corporation	12/7/1999
c:\winnt\system32\netui2.dll		
mprui.dll	5.00.2195.2104	54.77 KB (56,080 bytes)
10:26:21 AM	Microsoft Corporation	9/24/2001
c:\winnt\system32\mprui.dll		
urlmon.dll	5.00.3315.1000	441.27 KB (451,856 bytes)
10:26:37 AM	Microsoft Corporation	9/24/2001
c:\winnt\system32\urlmon.dll		

browsecl.dll	5.00.3315.2846	34.50 KB (35,328 bytes)		md5filt.dll	5.00.0984	32.77 KB (33,552 bytes)	9/24/2001 10:27:35 AM
9/24/2001 10:26:10 AM	Microsoft Corporation			AM	Microsoft Corporation		
c:\winnt\system32\browsecl.dll				c:\winnt\system32\inetsrv\md5filt.dll			
faxshell.dll	5.00.2134.1	8.27 KB (8,464 bytes)	12/7/1999 7:00:00 AM	gzip.dll	5.00.0984	30.27 KB (30,992 bytes)	9/24/2001 10:27:34 AM
Microsoft Corporation	c:\winnt\system32\faxshell.dll			AM	Microsoft Corporation	c:\winnt\system32\inetsrv\gzip.dll	
msacm32.dll	5.00.2134.1	65.27 KB (66,832 bytes)		compfilt.dll	5.00.0984	22.77 KB (23,312 bytes)	9/24/2001 10:27:33 AM
12/7/1999 7:00:00 AM	Microsoft Corporation			AM	Microsoft Corporation		
c:\winnt\system32\msacm32.dll				c:\winnt\system32\inetsrv\compfilt.dll			
avifil32.dll	5.00.2134.1	76.27 KB (78,096 bytes)	12/7/1999 7:00:00 AM	sspfilt.dll	5.00.0984	43.27 KB (44,304 bytes)	9/24/2001 10:27:35 AM
Microsoft Corporation	c:\winnt\system32\avifil32.dll			AM	Microsoft Corporation		
c:\winnt\system32\avifil32.dll				c:\winnt\system32\inetsrv\sspfilt.dll			
msvfw32.dll	5.00.2134.1	113.77 KB (116,496 bytes)		iscomlog.dll	5.00.0984	24.77 KB (25,360 bytes)	9/24/2001 10:27:35 AM
12/7/1999 7:00:00 AM	Microsoft Corporation			AM	Microsoft Corporation		
c:\winnt\system32\msvfw32.dll				c:\winnt\system32\inetsrv\iscomlog.dll			
docprop2.dll	5.00.2178.1	297.77 KB (304,912 bytes)		lonsint.dll	5.00.0984	11.77 KB (12,048 bytes)	9/24/2001 10:27:35 AM
12/7/1999 7:00:00 AM	Microsoft Corporation			AM	Microsoft Corporation		
c:\winnt\system32\docprop2.dll				c:\winnt\system32\inetsrv\lonsint.dll			
linkinfo.dll	5.00.2134.1	15.77 KB (16,144 bytes)	12/7/1999 7:00:00 AM	inetsloc.dll	5.00.0984	20.27 KB (20,752 bytes)	9/24/2001 10:26:17 AM
Microsoft Corporation	c:\winnt\system32\linkinfo.dll			AM	Microsoft Corporation	c:\winnt\system32\inetsloc.dll	
c:\winnt\system32\linkinfo.dll				iisfecnv.dll	5.00.0984	7.27 KB (7,440 bytes)	9/17/2001 10:05:27 AM
powrprof.dll	5.00.3103.1000	13.27 KB (13,584 bytes)		Microsoft Corporation	c:\winnt\system32\inetsrv\iisfecnv.dll		
9/24/2001 10:26:32 AM	Microsoft Corporation			isatq.dll	5.00.0984	60.27 KB (61,712 bytes)	9/24/2001 10:27:35 AM
c:\winnt\system32\powrprof.dll				AM	Microsoft Corporation	c:\winnt\system32\inetsrv\isatq.dll	
batmeter.dll	5.00.3103.1000	20.27 KB (20,752 bytes)		infocomm.dll	5.00.0984	238.27 KB (243,984 bytes)	9/24/2001 10:27:34 AM
9/24/2001 10:26:10 AM	Microsoft Corporation			AM	Microsoft Corporation		
c:\winnt\system32\batmeter.dll				c:\winnt\system32\inetsrv\infocomm.dll			
stobject.dll	5.00.2195.2780	79.27 KB (81,168 bytes)	9/24/2001 10:26:36 AM	w3svc.dll	5.00.0984	343.27 KB (351,504 bytes)	9/24/2001 10:27:36 AM
Microsoft Corporation	c:\winnt\system32\stobject.dll			AM	Microsoft Corporation		
c:\winnt\system32\stobject.dll				c:\winnt\system32\inetsrv\w3svc.dll			
webcheck.dll	5.00.3315.1000	251.77 KB (257,808 bytes)		security.dll	5.00.2154.1	5.77 KB (5,904 bytes)	12/7/1999 7:00:00 AM
9/24/2001 10:26:38 AM	Microsoft Corporation			Microsoft Corporation	c:\winnt\system32\security.dll		
c:\winnt\system32\webcheck.dll				svceext.dll	5.00.0984	39.77 KB (40,720 bytes)	9/24/2001 10:27:35 AM
msi.dll	1.11.2405.0	1.69 MB (1,767,184 bytes)	9/24/2001 10:26:23 AM	AM	Microsoft Corporation		
Microsoft Corporation	c:\winnt\system32\msi.dll			c:\winnt\system32\inetsrv\svceext.dll			
c:\winnt\system32\msi.dll				admexs.dll	5.00.0984	27.77 KB (28,432 bytes)	9/24/2001 10:27:33 AM
ntshrui.dll	5.00.2134.1	46.77 KB (47,888 bytes)	12/7/1999 7:00:00 AM	AM	Microsoft Corporation		
Microsoft Corporation	c:\winnt\system32\ntshrui.dll			c:\winnt\system32\inetsrv\admexs.dll			
c:\winnt\system32\ntshrui.dll				wamreg.dll	5.00.0984	45.77 KB (46,864 bytes)	9/24/2001 10:27:36 AM
mydocs.dll	5.00.2920.0000	55.77 KB (57,104 bytes)	12/7/1999 7:00:00 AM	AM	Microsoft Corporation		
Microsoft Corporation	c:\winnt\system32\mydocs.dll			c:\winnt\system32\inetsrv\wamreg.dll			
c:\winnt\system32\mydocs.dll				metadata.dll	5.00.0984	68.77 KB (70,416 bytes)	9/24/2001 10:27:35 AM
browseui.dll	5.00.3315.2846	788.77 KB (807,696 bytes)		AM	Microsoft Corporation		
9/24/2001 10:26:10 AM	Microsoft Corporation			c:\winnt\system32\inetsrv\metadata.dll			
c:\winnt\system32\browseui.dll				iismap.dll	5.00.0984	55.77 KB (57,104 bytes)	9/24/2001 10:26:17 AM
shdocvw.dll	5.00.3315.2879	1.05 MB (1,104,144 bytes)		AM	Microsoft Corporation	c:\winnt\system32\iismap.dll	
9/24/2001 10:26:34 AM	Microsoft Corporation			nsepm.dll	5.00.0984	43.27 KB (44,304 bytes)	9/24/2001 10:27:35 AM
c:\winnt\system32\shdocvw.dll				AM	Microsoft Corporation		
explorer.exe	5.00.3315.2846	237.27 KB (242,960 bytes)		c:\winnt\system32\inetsrv\nsepm.dll			
9/24/2001 10:26:39 AM	Microsoft Corporation			admwpox.dll	5.00.0984	31.77 KB (32,528 bytes)	9/17/2001 10:05:29 AM
AM	Microsoft Corporation			AM	Microsoft Corporation		
c:\winnt\explorer.exe				c:\winnt\system32\admwpox.dll			
tapisrv.dll	5.00.2195.2955	169.27 KB (173,328 bytes)	9/24/2001 10:26:36 AM	coadmin.dll	5.00.0984	39.27 KB (40,208 bytes)	9/24/2001 10:27:33 AM
Microsoft Corporation	c:\winnt\system32\tapisrv.dll			AM	Microsoft Corporation		
c:\winnt\system32\tapisrv.dll				c:\winnt\system32\inetsrv\coadmin.dll			
dfssvc.exe	5.00.2195.2841	88.27 KB (90,384 bytes)	9/24/2001 10:26:13 AM	iisadmin.dll	5.00.0984	15.27 KB (15,632 bytes)	9/24/2001 10:27:34 AM
Microsoft Corporation	c:\winnt\system32\dfssvc.exe			AM	Microsoft Corporation		
c:\winnt\system32\dfssvc.exe				c:\winnt\system32\inetsrv\iisadmin.dll			
iislog.dll	5.00.0984	75.27 KB (77,072 bytes)	9/24/2001 10:27:34 AM	rpref.dll	5.00.0984	4.27 KB (4,368 bytes)	9/24/2001 10:27:35 AM
AM	Microsoft Corporation	c:\winnt\system32\inetsrv\iislog.dll		Microsoft Corporation	c:\winnt\system32\inetsrv\rpref.dll		
c:\winnt\system32\inetsrv\iislog.dll				iisrtl.dll	5.00.0984	119.77 KB (122,640 bytes)	9/24/2001 10:26:17 AM
httpext.dll	0.9.3940.21	435.27 KB (445,712 bytes)	9/24/2001 10:27:34 AM	AM	Microsoft Corporation	c:\winnt\system32\iisrtl.dll	
Microsoft Corporation	c:\winnt\system32\inetsrv\httpext.dll			inetinfo.exe	5.00.0984	14.27 KB (14,608 bytes)	9/24/2001 10:27:34 AM
c:\winnt\system32\inetsrv\httpext.dll				AM	Microsoft Corporation		
rpcproxy.dll	5.00.2195.2780	16.27 KB (16,656 bytes)		c:\winnt\system32\inetsrv\inetinfo.exe			
9/24/2001 10:27:26 AM	Microsoft Corporation			netui1.dll	5.00.2134.1	210.27 KB (215,312 bytes)	12/7/1999 7:00:00 AM
AM	Microsoft Corporation	c:\program files\common files\microsoft shared\web server extensions\40\bin\fpexedll.dll		AM	Microsoft Corporation		
c:\winnt\system32\rpcproxy\rpcproxy.dll				c:\winnt\system32\netui1.dll			
fpexedll.dll	4.0.2.4324	20.06 KB (20,541 bytes)	9/24/2001 10:27:24 AM				
Microsoft Corporation							
c:\program files\common files\microsoft shared\web server extensions\40\bin\fpexedll.dll							

netui0.dll	5.00.2134.1	70.27 KB (71,952 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\netui0.dll
ntlanman.dll	5.00.2157.1	35.27 KB (36,112 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntlanman.dll
wshnetbs.dll	5.00.2134.1	7.77 KB (7,952 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\wshnetbs.dll
ntmarta.dll	5.00.2195.2862	98.77 KB (101,136 bytes)	9/24/2001	10:26:29 AM	Microsoft Corporation	c:\winnt\system32\ntmarta.dll
perfos.dll	5.00.2155.1	21.27 KB (21,776 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\perfos.dll
provthrd.dll	1.50.1085.0000	68.07 KB (69,708 bytes)	9/17/2001	2:09:22 PM	Microsoft Corporation	c:\winnt\system32\wbem\provthrd.dll
ntevt.dll	1.50.1085.0000	192.06 KB (196,669 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\wbem\ntevt.dll
psapi.dll	5.00.2134.1	28.27 KB (28,944 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\psapi.dll
framedyn.dll	1.50.1085.0000	164.05 KB (167,992 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\wbem\framedyn.dll
cimwin32.dll	1.50.1085.0038	1.02 MB (1,073,232 bytes)	9/24/2001	10:26:41 AM	Microsoft Corporation	c:\winnt\system32\wbem\cimwin32.dll
wbemsvcs.dll	1.50.1085.0007	40.07 KB (41,036 bytes)	9/24/2001	10:26:43 AM	Microsoft Corporation	c:\winnt\system32\wbem\wbemsvcs.dll
wbemess.dll	1.50.1085.0039	364.07 KB (372,804 bytes)	9/24/2001	10:26:43 AM	Microsoft Corporation	c:\winnt\system32\wbem\wbemess.dll
fastprox.dll	1.50.1085.0037	144.08 KB (147,536 bytes)	9/24/2001	10:26:42 AM	Microsoft Corporation	c:\winnt\system32\wbem\fastprox.dll
wbemcore.dll	1.50.1085.0036	628.07 KB (643,140 bytes)	9/24/2001	10:26:42 AM	Microsoft Corporation	c:\winnt\system32\wbem\wbemcore.dll
wbemcomm.dll	1.50.1085.0021	692.07 KB (708,675 bytes)	9/24/2001	10:26:42 AM	Microsoft Corporation	c:\winnt\system32\wbem\wbemcomm.dll
winmgmt.exe	1.50.1085.0029	192.08 KB (196,685 bytes)	9/24/2001	10:26:43 AM	Microsoft Corporation	c:\winnt\system32\wbem\winmgmt.exe
simptcp.dll	5.00.2134.1	19.27 KB (19,728 bytes)	9/17/2001	10:04:47 AM	Microsoft Corporation	c:\winnt\system32\simptcp.dll
tcpsvcs.exe	5.00.2134.1	24.77 KB (25,360 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\tcpsvcs.exe
msidle.dll	5.00.2920.0000	6.27 KB (6,416 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\msidle.dll
mstask.exe	4.71.2195.1	115.27 KB (118,032 bytes)	9/24/2001	10:26:26 AM	Microsoft Corporation	c:\winnt\system32\mstask.exe
regsvc.exe	5.00.2195.2104	65.27 KB (66,832 bytes)	9/24/2001	10:26:33 AM	Microsoft Corporation	c:\winnt\system32\regsvc.exe
nmssvc.exe	1.64.0.0	1012.00 KB (1,036,288 bytes)	10/8/2001	4:31:07 PM	Intel Corporation	c:\winnt\system32\nmssvc.exe
llsrpc.dll	5.00.2149.1	45.77 KB (46,864 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\llsrpc.dll
llsrv.exe	5.00.2195.2649	114.27 KB (117,008 bytes)	5/4/2001	12:05:02 PM	Microsoft Corporation	c:\winnt\system32\llsrv.exe
wmi.dll	5.00.2191.1	6.27 KB (6,416 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\wmi.dll
netshell.dll	5.00.2195.2779	457.27 KB (468,240 bytes)	9/24/2001	10:26:28 AM	Microsoft Corporation	c:\winnt\system32\netshell.dll
netman.dll	5.00.2195.2779	89.27 KB (91,408 bytes)	9/24/2001	10:26:28 AM	Microsoft Corporation	c:\winnt\system32\netman.dll
ntmsdba.dll	5.00.2195.2779	167.27 KB (171,280 bytes)	9/24/2001	10:26:29 AM	Microsoft Corporation	c:\winnt\system32\ntmsdba.dll
rasdlg.dll	5.00.2195.2671	514.27 KB (526,608 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\rasdlg.dll
netcfgx.dll	5.00.2195.2228	534.77 KB (547,600 bytes)	9/24/2001	10:26:28 AM	Microsoft Corporation	c:\winnt\system32\netcfgx.dll
rasmans.dll	5.00.2195.2728	147.27 KB (150,800 bytes)	9/24/2001	10:26:32 AM	Microsoft Corporation	c:\winnt\system32\rasmans.dll
sens.dll	5.00.2163.1	36.77 KB (37,648 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\sens.dll
ntmssvc.dll	5.00.2195.2779	391.27 KB (400,656 bytes)	9/24/2001	10:26:29 AM	Microsoft Corporation	c:\winnt\system32\ntmssvc.dll
iashlpr.dll	5.00.2184.1	33.27 KB (34,064 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\iashlpr.dll
iasacct.dll	5.00.2134.1	28.27 KB (28,944 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\iasacct.dll
iasuser.dll	5.00.2134.1	25.77 KB (26,384 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\iasuser.dll
iasnap.dll	5.00.2195.2104	58.77 KB (60,176 bytes)	9/24/2001	10:26:16 AM	Microsoft Corporation	c:\winnt\system32\iasnap.dll
iaspipe.dll	5.00.2134.1	41.77 KB (42,768 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\iaspipe.dll
expshr.dll	6.0.8540	370.27 KB (379,152 bytes)	9/24/2001	10:26:16 AM	Microsoft Corporation	c:\winnt\system32\expshr.dll
vbajet32.dll	6.1.8268	30.27 KB (30,992 bytes)	9/24/2001	10:26:37 AM	Microsoft Corporation	c:\winnt\system32\vbajet32.dll
msjtes40.dll	4.00.4229.0	236.27 KB (241,936 bytes)	9/24/2001	10:26:25 AM	Microsoft Corporation	c:\winnt\system32\msjtes40.dll
oledb32r.dll	2.62.7926.0	68.27 KB (69,904 bytes)	2/15/2002	1:26:21 PM	Microsoft Corporation	c:\program files\common files\system\ole db\oledb32r.dll
comdlg32.dll	5.00.3103.1000	236.77 KB (242,448 bytes)	12/7/1999	7:00:00 AM	Microsoft Corporation	c:\winnt\system32\comdlg32.dll
msdart.dll	2.62.7926.0	144.27 KB (147,728 bytes)	2/15/2002	1:26:19 PM	Microsoft Corporation	c:\winnt\system32\msdart.dll
oledb32.dll	2.62.7926.0	448.27 KB (459,024 bytes)	2/15/2002	1:26:21 PM	Microsoft Corporation	c:\program files\common files\system\ole db\oledb32.dll
msjint40.dll	4.00.2927.2	148.27 KB (151,824 bytes)	9/24/2001	10:26:25 AM	Microsoft Corporation	c:\winnt\system32\msjint40.dll

msjter40.dll	4.00.2927.2	52.27 KB (53,520 bytes)		inetpp.dll	5.00.2195.2842	65.27 KB (66,832 bytes)	9/24/2001
9/24/2001 10:26:25 AM	Microsoft Corporation			10:26:17 AM	Microsoft Corporation		
c:\winnt\system32\msjter40.dll				c:\winnt\system32\inetpp.dll			
mswstr10.dll	4.00.3829.2	600.27 KB (614,672 bytes)		win32spl.dll	5.00.2195.2780	92.27 KB (94,480 bytes)	
9/24/2001 10:26:27 AM	Microsoft Corporation			12/7/1999 7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\mswstr10.dll				c:\winnt\system32\win32spl.dll			
msjet40.dll	4.00.4431.3	1.43 MB (1,503,504 bytes)	9/24/2001	usbmon.dll	5.00.2195.2780	11.27 KB (11,536 bytes)	9/24/2001
10:26:25 AM	Microsoft Corporation			10:26:37 AM	Microsoft Corporation		
c:\winnt\system32\msjet40.dll				c:\winnt\system32\usbmon.dll			
msjetoledb40.dll	4.00.4331.4	340.27 KB (348,432 bytes)		tcpmon.dll	5.00.2195.2780	40.77 KB (41,744 bytes)	9/24/2001
9/24/2001 10:26:25 AM	Microsoft Corporation			10:26:36 AM	Microsoft Corporation		
c:\winnt\system32\msjetoledb40.dll				c:\winnt\system32\tcpmon.dll			
iasrad.dll	5.00.2139.1	94.27 KB (96,528 bytes)	12/7/1999	pjlmon.dll	5.00.2165.1	12.77 KB (13,072 bytes)	11/30/1999
7:00:00 AM	Microsoft Corporation			6:39:36 PM	Microsoft Corporation		
c:\winnt\system32\iasrad.dll				c:\winnt\system32\pjlmon.dll			
iasam.dll	5.00.2160.1	96.27 KB (98,576 bytes)	12/7/1999	cnbjmon.dll	5.00.2134.1	43.77 KB (44,816 bytes)	
7:00:00 AM	Microsoft Corporation			11/30/1999 6:38:48 PM	Microsoft Corporation		
c:\winnt\system32\iasam.dll				c:\winnt\system32\cnbjmon.dll			
iasads.dll	5.00.2134.1	73.77 KB (75,536 bytes)	12/7/1999	localspl.dll	5.00.2195.2793	246.77 KB (252,688 bytes)	12/7/1999
7:00:00 AM	Microsoft Corporation			7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\iasads.dll				c:\winnt\system32\localspl.dll			
iaspolicy.dll	5.00.2134.1	25.27 KB (25,872 bytes)		spoolss.dll	5.00.2161.1	61.77 KB (63,248 bytes)	9/17/2001
12/7/1999 7:00:00 AM	Microsoft Corporation			9:39:14 AM	Microsoft Corporation		
c:\winnt\system32\iaspolicy.dll				c:\winnt\system32\spoolss.dll			
iasvcs.dll	5.00.2195.2104	58.77 KB (60,176 bytes)	9/24/2001	spoolsv.exe	5.00.2161.1	43.77 KB (44,816 bytes)	
10:26:17 AM	Microsoft Corporation			9/17/2001 9:39:14 AM	Microsoft Corporation		
c:\winnt\system32\iasvcs.dll				c:\winnt\system32\spoolsv.exe			
iasdo.dll	5.00.2195.2104	261.77 KB (268,048 bytes)	9/24/2001	rpcss.dll	5.00.2195.2815	231.27 KB (236,816 bytes)	9/24/2001
10:26:16 AM	Microsoft Corporation			10:26:33 AM	Microsoft Corporation		
c:\winnt\system32\iasdo.dll				c:\winnt\system32\rpcss.dll			
ias.dll	5.00.2134.1	7.27 KB (7,440 bytes)	12/7/1999 7:00:00 AM	svchost.exe	5.00.2134.1	7.77 KB (7,952 bytes)	12/7/1999
Microsoft Corporation				7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\ias.dll				c:\winnt\system32\svchost.exe			
es.dll	2000.2.3471.1	222.27 KB (227,600 bytes)	9/24/2001	dssenh.dll	5.00.2195.2228	142.77 KB (146,192 bytes)	9/24/2001
10:26:15 AM	Microsoft Corporation			10:27:28 AM	Microsoft Corporation		
c:\winnt\system32\es.dll				c:\winnt\system32\dssenh.dll			
mtxoci.dll	2000.2.3471.1	101.77 KB (104,208 bytes)	9/24/2001	oakley.dll	5.00.2195.2785	378.77 KB (387,856 bytes)	9/24/2001
10:26:27 AM	Microsoft Corporation			10:26:30 AM	Microsoft Corporation		
c:\winnt\system32\mtxoci.dll				c:\winnt\system32\oakley.dll			
resutils.dll	5.00.2195.2787	39.77 KB (40,720 bytes)	9/24/2001	mfc42u.dll	6.00.8665.0	972.05 KB (995,384 bytes)	12/7/1999
10:26:33 AM	Microsoft Corporation			7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\resutils.dll				c:\winnt\system32\mfc42u.dll			
clusapi.dll	5.00.2195.2104	54.27 KB (55,568 bytes)	9/24/2001	polagent.dll	5.00.2183.1	108.27 KB (110,864 bytes)	
10:26:11 AM	Microsoft Corporation			12/7/1999 7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\clusapi.dll				c:\winnt\system32\polagent.dll			
msvcpx50.dll	5.00.7051	552.50 KB (565,760 bytes)	12/7/1999	scecli.dll	5.00.2195.2780	105.27 KB (107,792 bytes)	9/24/2001
7:00:00 AM	Microsoft Corporation			10:26:33 AM	Microsoft Corporation		
c:\winnt\system32\msvcpx50.dll				c:\winnt\system32\scecli.dll			
xolehlp.dll	1999.9.3421.3	17.27 KB (17,680 bytes)	9/17/2001	atl.dll	3.00.8449	57.56 KB (58,938 bytes)	12/7/1999 7:00:00 AM
10:04:53 AM	Microsoft Corporation			Microsoft Corporation			
c:\winnt\system32\xolehlp.dll				c:\winnt\system32\atl.dll			
msdtclog.dll	1999.9.3421.3	89.77 KB (91,920 bytes)		certcli.dll	5.00.2195.2778	130.77 KB (133,904 bytes)	9/24/2001
9/17/2001 10:04:52 AM	Microsoft Corporation			10:26:11 AM	Microsoft Corporation		
c:\winnt\system32\msdtclog.dll				c:\winnt\system32\certcli.dll			
mtxclu.dll	2000.2.3471.1	51.27 KB (52,496 bytes)	9/24/2001	esent.dll	6.0.3940.13	1.08 MB (1,135,376 bytes)	9/24/2001
10:26:27 AM	Microsoft Corporation			10:26:15 AM	Microsoft Corporation		
c:\winnt\system32\mtxclu.dll				c:\winnt\system32\esent.dll			
msdtcprx.dll	2000.2.3471.1	665.77 KB (681,744 bytes)		ntdsatq.dll	5.00.2195.2878	31.27 KB (32,016 bytes)	9/24/2001
9/24/2001 10:26:21 AM	Microsoft Corporation			10:26:29 AM	Microsoft Corporation		
c:\winnt\system32\msdtcprx.dll				c:\winnt\system32\ntdsatq.dll			
txfaux.dll	2000.2.3471.1	374.27 KB (383,248 bytes)	9/24/2001	ntdsa.dll	5.00.2195.2899	990.77 KB (1,014,544 bytes)	9/24/2001
10:26:37 AM	Microsoft Corporation			10:26:29 AM	Microsoft Corporation		
c:\winnt\system32\txfaux.dll				c:\winnt\system32\ntdsa.dll			
msdtctm.dll	2000.2.3471.1	1.07 MB (1,120,528 bytes)		kdcsvc.dll	5.00.2195.2878	137.77 KB (141,072 bytes)	9/24/2001
9/24/2001 10:26:21 AM	Microsoft Corporation			10:26:19 AM	Microsoft Corporation		
c:\winnt\system32\msdtctm.dll				c:\winnt\system32\kdcsvc.dll			
msdtc.exe	1999.9.3421.3	6.77 KB (6,928 bytes)	9/17/2001 10:04:52 AM	sfnapi.dll	5.00.2134.1	38.77 KB (39,696 bytes)	12/7/1999
Microsoft Corporation				7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\msdtc.exe				c:\winnt\system32\sfnapi.dll			

rassfm.dll 5.00.2195.2671 21.27 KB (21,776 bytes) 9/24/2001 10:26:32 AM Microsoft Corporation	trkwns.dll 5.00.2166.1 88.77 KB (90,896 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\rassfm.dll	c:\winnt\system32\trkwns.dll
mpr.dll 5.00.2195.2779 53.27 KB (54,544 bytes) 9/24/2001 10:26:20 AM Microsoft Corporation	seclogon.dll 5.00.2135.1 15.77 KB (16,144 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\mpr.dll	c:\winnt\system32\seclogon.dll
rsabase.dll 5.00.2195.2228 128.27 KB (131,344 bytes) 5/4/2001 12:05:02 PM Microsoft Corporation	psbase.dll 5.00.2195.2779 111.77 KB (114,448 bytes) 9/24/2001 10:26:32 AM Microsoft Corporation
c:\winnt\system32\rsabase.dll	c:\winnt\system32\psbase.dll
schannel.dll 5.00.2195.2922 138.27 KB (141,584 bytes) 5/4/2001 12:05:02 PM Microsoft Corporation	cryptsvc.dll 5.00.2181.1 61.77 KB (63,248 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\schannel.dll	c:\winnt\system32\cryptsvc.dll
netlogon.dll 5.00.2195.2865 357.77 KB (366,352 bytes) 9/24/2001 10:26:28 AM Microsoft Corporation	cryptdll.dll 5.00.2135.1 41.27 KB (42,256 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\netlogon.dll	c:\winnt\system32\cryptdll.dll
msv1_0.dll 5.00.2195.2900 111.77 KB (114,448 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation	wkssvc.dll 5.00.2195.2780 95.27 KB (97,552 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\msv1_0.dll	c:\winnt\system32\wkssvc.dll
kerberos.dll 5.00.2195.2913 198.77 KB (203,536 bytes) 9/24/2001 10:26:19 AM Microsoft Corporation	srsvcs.dll 5.00.2195.2904 79.27 KB (81,168 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\kerberos.dll	c:\winnt\system32\srsvcs.dll
msprvs.dll 5.00.2154.1 41.50 KB (42,496 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation	cfgmgr32.dll 5.00.2134.1 16.77 KB (17,168 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\msprvs.dll	c:\winnt\system32\cfgmgr32.dll
samsrv.dll 5.00.2195.2918 369.77 KB (378,640 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation	dmserver.dll 2195.2778.297.3 11.77 KB (12,048 bytes) 9/24/2001 10:26:14 AM VERITAS Software Corp.
c:\winnt\system32\samsrv.dll	c:\winnt\system32\dmserver.dll
lsasrv.dll 5.00.2195.2964 492.77 KB (504,592 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation	winsta.dll 5.00.2195.2386 36.77 KB (37,648 bytes) 9/24/2001 10:26:38 AM Microsoft Corporation
c:\winnt\system32\lsasrv.dll	c:\winnt\system32\winsta.dll
lsass.exe 5.00.2195.2964 32.77 KB (33,552 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation	lmhsvc.dll 5.00.2195.2778 9.77 KB (10,000 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\lsass.exe	c:\winnt\system32\lmhsvc.dll
ntlsapi.dll 5.00.2134.1 6.77 KB (6,928 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation	dnssrslvr.dll 5.00.2195.2778 88.77 KB (90,896 bytes) 9/24/2001 10:26:14 AM Microsoft Corporation
c:\winnt\system32\ntlsapi.dll	c:\winnt\system32\dnssrslvr.dll
wmicore.dll 5.00.2195.2842 72.27 KB (74,000 bytes) 9/24/2001 10:26:39 AM Microsoft Corporation	clbcatq.dll 2000.2.3471.1 496.77 KB (508,688 bytes) 9/24/2001 10:26:11 AM Microsoft Corporation
c:\winnt\system32\wmicore.dll	c:\winnt\system32\clbcatq.dll
rasadhlp.dll 5.00.2168.1 7.27 KB (7,440 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation	tapi32.dll 5.00.2182.1 123.27 KB (126,224 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\rasadhlp.dll	c:\winnt\system32\tapi32.dll
winrnr.dll 5.00.2160.1 18.77 KB (19,216 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation	rasman.dll 5.00.2195.2780 54.77 KB (56,080 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\winrnr.dll	c:\winnt\system32\rasman.dll
rnr20.dll 5.00.2195.2871 35.77 KB (36,624 bytes) 9/24/2001 10:26:33 AM Microsoft Corporation	rasapi32.dll 5.00.2195.2671 189.77 KB (194,320 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\rnr20.dll	c:\winnt\system32\rasapi32.dll
wshtcpip.dll 5.00.2195.2104 17.27 KB (17,680 bytes) 9/24/2001 10:26:39 AM Microsoft Corporation	rtutils.dll 5.00.2168.1 43.77 KB (44,816 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\wshtcpip.dll	c:\winnt\system32\rtutils.dll
msafld.dll 5.00.2195.2779 106.77 KB (109,328 bytes) 9/24/2001 10:26:21 AM Microsoft Corporation	adslpdc.dll 5.00.2195.2842 127.27 KB (130,320 bytes) 9/24/2001 10:26:09 AM Microsoft Corporation
c:\winnt\system32\msafld.dll	c:\winnt\system32\adslpdc.dll
mswsock.dll 5.00.2195.2871 62.77 KB (64,272 bytes) 9/24/2001 10:26:27 AM Microsoft Corporation	activeds.dll 5.00.2195.2778 174.77 KB (178,960 bytes) 9/24/2001 10:26:02 AM Microsoft Corporation
c:\winnt\system32\mswsock.dll	c:\winnt\system32\activeds.dll
xactsrv.dll 5.00.2134.1 90.27 KB (92,432 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation	oleaut32.dll 2.40.4517 612.27 KB (626,960 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\xactsrv.dll	c:\winnt\system32\oleaut32.dll
msgsvc.dll 5.00.2195.2939 34.27 KB (35,088 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation	mprapi.dll 5.00.2181.1 79.27 KB (81,168 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\msgsvc.dll	c:\winnt\system32\mprapi.dll
browser.dll 5.00.2195.2778 48.27 KB (49,424 bytes) 9/24/2001 10:26:10 AM Microsoft Corporation	iphlpapi.dll 5.00.2173.2 67.77 KB (69,392 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\browser.dll	c:\winnt\system32\iphlpapi.dll
alrsvc.dll 5.00.2134.1 17.77 KB (18,192 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation	icmp.dll 5.00.2134.1 7.27 KB (7,440 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation
c:\winnt\system32\alrsvc.dll	c:\winnt\system32\icmp.dll

dhcpcsvc.dll	5.00.2195.2778	88.77 KB (90,896 bytes)		comctl32.dll	5.81	537.77 KB (550,672 bytes)	12/7/1999
12/7/1999 7:00:00 AM	Microsoft Corporation			7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\dhcpcsvc.dll				c:\winnt\system32\comctl32.dll			
eventlog.dll	5.00.2178.1	43.77 KB (44,816 bytes)		shlwapi.dll	5.00.3315.1000	282.77 KB (289,552 bytes)	9/24/2001
12/7/1999 7:00:00 AM	Microsoft Corporation			10:26:35 AM	Microsoft Corporation		
c:\winnt\system32\eventlog.dll				c:\winnt\system32\shlwapi.dll			
ntdsapi.dll	5.00.2195.2661	55.77 KB (57,104 bytes)	9/24/2001	shell32.dll	5.00.3315.2902	2.25 MB (2,359,056 bytes)	9/24/2001
10:26:29 AM	Microsoft Corporation			10:26:35 AM	Microsoft Corporation		
c:\winnt\system32\ntdsapi.dll				c:\winnt\system32\shell32.dll			
scsersv.dll	5.00.2195.2780	226.27 KB (231,696 bytes)	9/24/2001	msgina.dll	5.00.2195.2779	324.27 KB (332,048 bytes)	12/7/1999
10:26:33 AM	Microsoft Corporation			7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\scsersv.dll				c:\winnt\system32\msgina.dll			
umpnpgmgr.dll	5.00.2182.1	86.27 KB (88,336 bytes)		wsock32.dll	5.00.2195.2871	21.27 KB (21,776 bytes)	
12/7/1999 7:00:00 AM	Microsoft Corporation			9/24/2001 10:26:39 AM	Microsoft Corporation		
c:\winnt\system32\umpnpgmgr.dll				c:\winnt\system32\wsock32.dll			
services.exe	5.00.2195.2780	86.77 KB (88,848 bytes)		dnsapi.dll	5.00.2195.2785	130.77 KB (133,904 bytes)	9/24/2001
12/7/1999 7:00:00 AM	Microsoft Corporation			10:26:14 AM	Microsoft Corporation		
c:\winnt\system32\services.exe				c:\winnt\system32\dnsapi.dll			
cscui.dll	5.00.2195.2959	228.27 KB (233,744 bytes)	9/24/2001	wldap32.dll	5.00.2195.2797	125.27 KB (128,272 bytes)	
10:26:12 AM	Microsoft Corporation			9/24/2001 10:26:38 AM	Microsoft Corporation		
c:\winnt\system32\cscui.dll				c:\winnt\system32\wldap32.dll			
winspool.drv	5.00.2195.2780	109.77 KB (112,400 bytes)		ws2help.dll	5.00.2134.1	17.77 KB (18,192 bytes)	
12/7/1999 7:00:00 AM	Microsoft Corporation			12/7/1999 7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\winspool.drv				c:\winnt\system32\ws2help.dll			
winscard.dll	5.00.2134.1	77.27 KB (79,120 bytes)		ws2_32.dll	5.00.2195.2780	67.77 KB (69,392 bytes)	9/24/2001
12/7/1999 7:00:00 AM	Microsoft Corporation			10:26:39 AM	Microsoft Corporation		
c:\winnt\system32\winscard.dll				c:\winnt\system32\ws2_32.dll			
wlnotify.dll	5.00.2195.2780	53.77 KB (55,056 bytes)		samlib.dll	5.00.2195.2780	49.77 KB (50,960 bytes)	12/7/1999
9/24/2001 10:26:39 AM	Microsoft Corporation			7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\wlnotify.dll				c:\winnt\system32\samlib.dll			
csd.dll	5.00.2195.2401	98.27 KB (100,624 bytes)	9/24/2001	netrap.dll	5.00.2134.1	11.27 KB (11,536 bytes)	12/7/1999
10:26:12 AM	Microsoft Corporation			7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\csd.dll				c:\winnt\system32\netrap.dll			
lz32.dll	5.00.2134.1	9.77 KB (10,000 bytes)	12/7/1999	netapi32.dll	5.00.2195.2808	303.77 KB (311,056 bytes)	
7:00:00 AM	Microsoft Corporation			9/24/2001 10:26:28 AM	Microsoft Corporation		
c:\winnt\system32\lz32.dll				c:\winnt\system32\netapi32.dll			
version.dll	5.00.2134.1	15.77 KB (16,144 bytes)	12/7/1999	profmap.dll	5.00.2181.1	29.27 KB (29,968 bytes)	
7:00:00 AM	Microsoft Corporation			12/7/1999 7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\version.dll				c:\winnt\system32\profmap.dll			
rsaenh.dll	5.00.2195.2228	130.77 KB (133,904 bytes)	9/24/2001	secur32.dll	5.00.2195.2862	46.77 KB (47,888 bytes)	9/24/2001
10:27:28 AM	Microsoft Corporation			10:26:34 AM	Microsoft Corporation		
c:\winnt\system32\rsaenh.dll				c:\winnt\system32\secur32.dll			
mcat32.dll	5.131.2134.1	7.77 KB (7,952 bytes)	12/7/1999	sfc.dll	5.00.2195.2896	92.11 KB (94,320 bytes)	9/24/2001
7:00:00 AM	Microsoft Corporation			10:26:34 AM	Microsoft Corporation		
c:\winnt\system32\mcat32.dll				c:\winnt\system32\sfc.dll			
ole32.dll	5.00.2195.2887	969.77 KB (993,040 bytes)	9/24/2001	nddeapi.dll	5.00.2137.1	15.27 KB (15,632 bytes)	12/7/1999
10:26:31 AM	Microsoft Corporation			7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\ole32.dll				c:\winnt\system32\nddeapi.dll			
imagehlp.dll	5.00.2195.2778	125.77 KB (128,784 bytes)		userenv.dll	5.00.2195.2780	361.77 KB (370,448 bytes)	12/7/1999
5/4/2001 12:05:02 PM	Microsoft Corporation			7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\imagehlp.dll				c:\winnt\system32\userenv.dll			
msasn1.dll	5.00.2134.1	51.27 KB (52,496 bytes)	12/7/1999	user32.dll	5.00.2195.2821	392.77 KB (402,192 bytes)	12/7/1999
7:00:00 AM	Microsoft Corporation			7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\msasn1.dll				c:\winnt\system32\user32.dll			
crypt32.dll	5.131.2195.2833	451.27 KB (462,096 bytes)	9/24/2001	gdi32.dll	5.00.2195.2778	228.77 KB (234,256 bytes)	12/7/1999
10:26:12 AM	Microsoft Corporation			7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\crypt32.dll				c:\winnt\system32\gdi32.dll			
wintrust.dll	5.131.2195.2779	162.27 KB (166,160 bytes)		rpert4.dll	5.00.2195.2832	437.27 KB (447,760 bytes)	9/24/2001
9/24/2001 10:26:38 AM	Microsoft Corporation			10:26:33 AM	Microsoft Corporation		
c:\winnt\system32\wintrust.dll				c:\winnt\system32\rpert4.dll			
setupapi.dll	5.00.2195.2663	555.77 KB (569,104 bytes)		advapi32.dll	5.00.2195.2867	351.77 KB (360,208 bytes)	
12/7/1999 7:00:00 AM	Microsoft Corporation			12/7/1999 7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\setupapi.dll				c:\winnt\system32\advapi32.dll			
winmm.dll	5.00.2161.1	184.77 KB (189,200 bytes)	12/7/1999	kernel32.dll	5.00.2195.2778	714.77 KB (731,920 bytes)	
7:00:00 AM	Microsoft Corporation			12/7/1999 7:00:00 AM	Microsoft Corporation		
c:\winnt\system32\winmm.dll				c:\winnt\system32\kernel32.dll			

msvcrt.dll 6.10.8924.0 284.05 KB (290,869 bytes) 5/4/2001
 12:05:02 PM Microsoft Corporation
 c:\winnt\system32\msvcrt.dll
 winlogon.exe 5.00.2195.2953 173.77 KB (177,936 bytes)
 12/7/1999 7:00:00 AM Microsoft Corporation
 c:\winnt\system32\winlogon.exe
 sfcfiles.dll 5.00.2195.2967 948.27 KB (971,024 bytes) 9/24/2001
 10:26:34 AM Microsoft Corporation
 c:\winnt\system32\sfcfiles.dll
 ntdll.dll 5.00.2195.2779 478.77 KB (490,256 bytes) 5/4/2001
 12:05:02 PM Microsoft Corporation
 c:\winnt\system32\ntdll.dll
 smss.exe 5.00.2195.2901 44.27 KB (45,328 bytes) 12/7/1999
 7:00:00 AM Microsoft Corporation
 c:\winnt\system32\smss.exe

[Services]

Display Name	Name	State	Start Mode	Service Type
Path	Error Control	Start Name	Tag ID	
Alert	Alert	Running	Auto	Share Process
c:\winnt\system32\services.exe		Normal	LocalSystem	0
Application Management	AppMgmt	Stopped	Manual	Share
Process c:\winnt\system32\services.exe		Normal	LocalSystem	0
Computer Browser	Browser	Running	Auto	Share Process
c:\winnt\system32\services.exe		Normal	LocalSystem	0
Indexing Service	cisvc	Stopped	Manual	Share Process
c:\winnt\system32\cisvc.exe		Normal	LocalSystem	0
ClipBook	ClipSrv	Stopped	Manual	Own Process
c:\winnt\system32\clipsrv.exe		Normal	LocalSystem	0
Distributed File System	Dfs	Running	Auto	Own
Process c:\winnt\system32\dfsrv.exe		Normal	LocalSystem	0
DHCP Client	Dhcp	Running	Auto	Share Process
c:\winnt\system32\services.exe		Normal	LocalSystem	0
Logical Disk Manager	Administrative Service		dmadmin	Stopped
Manual Share Process			c:\winnt\system32\dmadmin.exe /com	
Normal LocalSystem			0	
Logical Disk Manager	dmserver	Running	Auto	Share Process
c:\winnt\system32\services.exe		Normal	LocalSystem	0
DNS Client	Dnscache	Running	Auto	Share Process
c:\winnt\system32\services.exe		Normal	LocalSystem	0
Event Log	Eventlog	Running	Auto	Share Process
c:\winnt\system32\services.exe		Normal	LocalSystem	0
COM+ Event System	EventSystem	Running	Manual	Share
Process c:\winnt\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0
Fax Service	Fax	Stopped	Manual	Own Process
c:\winnt\system32\faxsvc.exe		Normal	LocalSystem	0
Internet Authentication Service	IAS	Running	Auto	Share
Process c:\winnt\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0
IIS Admin Service	IISADMIN	Running	Auto	Share
Process c:\winnt\system32\inetrv\inetinfo.exe		Normal	LocalSystem	0
Intersite Messaging	IsmServ	Stopped	Disabled	Own Process
c:\winnt\system32\ismserv.exe		Normal	LocalSystem	0
Kerberos Key Distribution Center	kdc	Stopped	Disabled	Share Process
Share Process c:\winnt\system32\lsass.exe		Normal	LocalSystem	0
Server	lanmanserver	Running	Auto	Share Process
c:\winnt\system32\services.exe		Normal	LocalSystem	0
Workstation	lanmanworkstation	Running	Auto	Share
Process c:\winnt\system32\services.exe		Normal	LocalSystem	0
License Logging Service	LicenseService	Running	Auto	Own Process
Own Process c:\winnt\system32\llssrv.exe		Normal	LocalSystem	0
TCP/IP NetBIOS Helper Service	LmHosts	Running	Auto	Share
Process c:\winnt\system32\services.exe		Normal	LocalSystem	0

Messenger Messenger	Running	Auto	Share Process
c:\winnt\system32\services.exe	Normal	LocalSystem	0
NetMeeting Remote Desktop Sharing	mnmsrv	Stopped	Manual
Own Process c:\winnt\system32\mnmsrv.exe		Normal	LocalSystem
Distributed Transaction Coordinator		MSDTC	Running
Own Process c:\winnt\system32\msdtc.exe		Normal	LocalSystem
Windows Installer	MSIServer	Stopped	Manual
c:\winnt\system32\msiexec.exe /v		Normal	LocalSystem
Network DDE	NetDDE	Stopped	Manual
c:\winnt\system32\netdde.exe		Normal	LocalSystem
Network DDE DSDM	NetDDEdsdm	Stopped	Manual
Process c:\winnt\system32\netdde.exe		Normal	LocalSystem
Net Logon	Netlogon	Stopped	Manual
c:\winnt\system32\lsass.exe		Normal	LocalSystem
Network Connections	Netman	Running	Manual
c:\winnt\system32\svchost.exe -k netsvcs		Normal	LocalSystem
NMS Service	NMSSvc	Running	Auto
c:\winnt\system32\nmssvc.exe		Normal	LocalSystem
File Replication	NtFrs	Stopped	Manual
c:\winnt\system32\ntfrs.exe		Ignore	LocalSystem
NT LM Security Support Provider		NtLmSsp	Stopped
Share Process c:\winnt\system32\lsass.exe		Normal	LocalSystem
Removable Storage	NtmsSvc	Running	Auto
c:\winnt\system32\svchost.exe -k netsvcs		Normal	LocalSystem
Plug and Play	PlugPlay	Running	Auto
c:\winnt\system32\services.exe		Normal	LocalSystem
IPSEC Policy Agent	PolicyAgent	Running	Auto
Process c:\winnt\system32\lsass.exe		Normal	LocalSystem
Protected Storage	ProtectedStorage	Running	Auto
Process c:\winnt\system32\services.exe		Normal	LocalSystem
Remote Access Auto Connection Manager	RasAuto	Stopped	Manual
Share Process c:\winnt\system32\svchost.exe -k netsvcs		Normal	LocalSystem
Remote Access Connection Manager		RasMan	Stopped
Share Process c:\winnt\system32\svchost.exe -k netsvcs		Normal	LocalSystem
Routing and Remote Access	RemoteAccess	Stopped	Disabled
Share Process c:\winnt\system32\svchost.exe -k netsvcs		Normal	LocalSystem
Remote Registry Service	RemoteRegistry	Running	Auto
Own Process c:\winnt\system32\regsvc.exe		Normal	LocalSystem
Remote Procedure Call (RPC) Locator		RpcLocator	Stopped
Manual Own Process c:\winnt\system32\locator.exe		Normal	LocalSystem
Remote Procedure Call (RPC)		RpcSs	Running
Process c:\winnt\system32\svchost.exe -k rps		Normal	LocalSystem
QoS Admission Control (RSVP)	RSVP	Running	Auto
Process c:\winnt\system32\rsrv.exe -s		Normal	LocalSystem
Security Accounts Manager	SamSs	Running	Auto
Process c:\winnt\system32\lsass.exe		Normal	LocalSystem
Smart Card Helper	SCardDrv	Stopped	Manual
c:\winnt\system32\scardsvr.exe		Ignore	LocalSystem
Smart Card	SCardSvr	Stopped	Manual
c:\winnt\system32\scardsvr.exe		Ignore	LocalSystem
Task Scheduler	Schedule	Running	Auto
c:\winnt\system32\nttask.exe		Normal	LocalSystem
RunAs Service	seclogon	Running	Auto
c:\winnt\system32\services.exe		Ignore	LocalSystem
System Event Notification	SENS	Running	Auto
Process c:\winnt\system32\svchost.exe -k netsvcs		Normal	LocalSystem

Internet Connection Sharing	SharedAccess	Stopped	Manual
Share Process	c:\winnt\system32\svchost.exe -k netsvcs		Normal
LocalSystem	0		
Simple TCP/IP Services	SimpTcp	Running	Auto
Process	c:\winnt\system32\tcpvcs.exe	Normal	LocalSystem
Print Spooler	Spooler	Running	Auto
c:\winnt\system32\spoolsv.exe	Normal	LocalSystem	0
Performance Logs and Alerts	SysmonLog	Stopped	Manual
Own Process	c:\winnt\system32\smlogsvc.exe	Normal	
LocalSystem	0		
Telephony TapiSrv	Running	Manual	Share Process
c:\winnt\system32\svchost.exe -k tapisrv	Normal	LocalSystem	0
Terminal Services	TermService	Stopped	Disabled
Process	c:\winnt\system32\termsrv.exe	Normal	LocalSystem
Telnet	TlntSvr	Stopped	Manual
c:\winnt\system32\tlntsvr.exe	Normal	LocalSystem	0
Distributed Link Tracking Server	TrkSvr	Stopped	Manual
Process	c:\winnt\system32\services.exe	Normal	LocalSystem
Distributed Link Tracking Client	TrkWks	Running	Auto
Process	c:\winnt\system32\services.exe	Normal	LocalSystem
Uninterruptible Power Supply	UPS	Stopped	Manual
Process	c:\winnt\system32\ups.exe	Normal	LocalSystem
Utility Manager	UtilMan	Stopped	Manual
c:\winnt\system32\utilman.exe	Normal	LocalSystem	0
Windows Time	W32Time	Stopped	Manual
c:\winnt\system32\services.exe	Normal	LocalSystem	0
World Wide Web Publishing Service	W3SVC	Running	Auto
Share Process	c:\winnt\system32\inetrv\inetinfo.exe	Normal	
LocalSystem	0		
Windows Management Instrumentation	WinMgmt	Running	Auto
Own Process	c:\winnt\system32\wbem\winmgmt.exe		Ignore
LocalSystem	0		
Windows Management Instrumentation Driver Extensions			Wmi
Running	Manual	Share Process	c:\winnt\system32\services.exe
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		
Accessories\System Tools	Default User:Accessories\System Tools	
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\Games	All Users:Accessories\Games	All Users
Accessories\Microsoft Script Debugger	All Users:Accessories\Microsoft	
Script Debugger	All Users	
Accessories\System Tools	All Users:Accessories\System Tools	
All Users		
Administrative Tools	All Users:Administrative Tools	All Users
Microsoft SQL Server	All Users:Microsoft SQL Server	All Users
Startup	All Users:Startup	All Users
Accessories	CLIENT10\Administrator:Accessories	
CLIENT10\Administrator		
Accessories\Accessibility		
CLIENT10\Administrator:Accessories\Accessibility		
CLIENT10\Administrator		

Accessories\Entertainment	
CLIENT10\Administrator:Accessories\Entertainment	
CLIENT10\Administrator	
Accessories\System Tools	CLIENT10\Administrator:Accessories\System
Tools	CLIENT10\Administrator
Administrative Tools	CLIENT10\Administrator:Administrative Tools
CLIENT10\Administrator	
Startup	CLIENT10\Administrator:Startup
CLIENT10\Administrator	

[Startup Programs]

Program	Command	User Name	Location
Promon.exe		promon.exe	All Users
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run			

[OLE Registration]

Object	Local Server
Sound (OLE2)	sndrec32.exe
Media Clip	mplay32.exe
Video Clip	mplay32.exe /avi
MIDI Sequence	mplay32.exe /mid
Sound	Not Available
Media Clip	Not Available
Image Document	"C:\Program Files\Windows NT\Accessories\ImageVue\KodakImg.exe"
WordPad Document	"%ProgramFiles%\Windows NT\Accessories\WORDPAD.EXE"
Windows Media Services DRM Storage object	Not Available
Bitmap Image	C:\WINNT\System32\mspaint.exe

[Internet Explorer 5]

[Following are sub-categories of this main category]

[Summary]

Item	Value
Version	5.00.3315.1000
Build	53315.1000
Product ID	51876-270-7339724-05107
Application Path	C:\Program Files\Internet Explorer
Language	English (United States)
Active Printer	Not Available
Cipher Strength	168-bit
Content Advisor	Disabled
IEAK Install	No

[File Versions]

File	Version	Size	Date	Path	Company
advapi32.dll	5.0.2195.2867	352 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation
advpack.dll	5.0.3103.1000	87 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation
browseui.dll	5.0.3315.2846	35 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation
browseui.dll	5.0.3315.2846	789 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation
ckenv.exe	5.0.2189.1	9 KB	12/7/1999 8:00:00 AM	C:\WINNT\system32	Microsoft Corporation
comctl32.dll	5.81.3103.1000	538 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation
crypt32.dll	5.131.2195.2833	451 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation

enhsg.dll	<File Missing>	Not Available	Not Available
Not Available	Not Available		
iemigrat.dll	<File Missing>	Not Available	Not
Available	Not Available	Not Available	
iesetup.dll	5.0.3103.1000	57 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
iexplore.exe	5.0.2920.0	59 KB	12/7/1999 8:00:00 AM
C:\Program Files\Internet Explorer	Microsoft Corporation		
imagehlp.dll	5.0.2195.2778	126 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
imgghelp.dll	<File Missing>	Not Available	Not
Available	Not Available	Not Available	
inseng.dll	5.0.3103.1000	72 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
jobexec.dll	5.0.0.1	47 KB	12/7/1999 8:00:00 AM
C:\WINNT\system32	Microsoft Corporation		
jscrip.dll	5.1.0.5907	476 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
jsproxy.dll	5.0.2920.0	13 KB	12/7/1999 8:00:00 AM
C:\WINNT\system32	Microsoft Corporation		
msahtml.dll	<File Missing>	Not Available	Not
Available	Not Available	Not Available	
mshhtml.dll	5.0.3315.2870	2290 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
msjava.dll	5.0.3802.0	923 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
msoss.dll	<File Missing>	Not Available	Not Available
Not Available	Not Available		
msxml.dll	8.0.5718.1	493 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
occache.dll	5.0.3103.1000	86 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
ole32.dll	5.0.2195.2887	970 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
oleaut32.dll	2.40.4517.0	612 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
olepro32.dll	5.0.4517.0	160 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
rsabase.dll	5.0.2195.2228	128 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
rsaenh.dll	5.0.2195.2228	131 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
rsapi32.dll	<File Missing>	Not Available	Not Available
Not Available	Not Available		
rsasig.dll	<File Missing>	Not Available	Not Available
Not Available	Not Available		
schannel.dll	5.1.2195.0	138 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
shdoc401.dll	<File Missing>	Not Available	Not
Available	Not Available	Not Available	
shdocvw.dll	5.0.3315.2879	1078 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
shell32.dll	5.0.3315.2902	2304 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
shlwapi.dll	5.0.3315.1000	283 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
url.dll	5.0.2920.0	82 KB	12/7/1999 8:00:00 AM
C:\WINNT\system32	Microsoft Corporation		
urlmon.dll	5.0.3315.1000	441 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
vbscript.dll	5.1.0.5907	428 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
webcheck.dll	5.0.3315.1000	252 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
win.com	5.0.2134.1	24 KB	12/7/1999 8:00:00 AM
C:\WINNT\system32	Microsoft Corporation		
wininet.dll	5.0.3315.1000	457 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		

winsock.dll	3.10.0.103	3 KB	12/7/1999 8:00:00 AM
C:\WINNT\system32	Microsoft Corporation		
wintrust.dll	5.131.2195.2779	162 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
wsock.vxd	<File Missing>	Not Available	Not Available
Not Available	Not Available		
wsock32.dll	5.0.2195.2871	21 KB	5/4/2001 12:05:02 PM
C:\WINNT\system32	Microsoft Corporation		
wsock32n.dll	<File Missing>	Not Available	Not
Available	Not Available	Not Available	

[Connectivity]

Item	Value
Connection Preference	Never dial
EnableHttp1.1	1
ProxyHttp1.1	0

LAN Settings

AutoConfigProxy	wininet.dll
AutoProxyDetectMode	Enabled
AutoConfigURL	
Proxy	Disabled
ProxyServer	
ProxyOverride	

[Cache]

[Following are sub-categories of this main category]

[Summary]

Item	Value
Page Refresh Type	Automatic
Temporary Internet Files Folder	C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files
Total Disk Space	8667 MB
Available Disk Space	5733 MB
Maximum Cache Size	270 MB
Available Cache Size	271 MB

[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
Administrator	Administrator	9/17/2001 to 8/24/2101	sha1RSA	

[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
Local intranet	Medium-low
Trusted sites	Low
Internet	Medium
Restricted sites	High

COM+ Settings

Client Configuration Parameters

COM+ Settings

TPCC.AllTxns:

Activation:

- Enable Object Pooling selected
- Minimum Pool Size: 187
- Maximum Pool Size: 187
- Creating Timeout: 60,000
- Enable Object Construction
- Enable Just in Time Activation

Concurrency:

- Concurrency Required

TPCC Application Registry Parameters

TPCC Application Registry Parameters

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPCC]
"Path"="c:\inetpub\wwwroot\"
"NumberOfDeliveryThreads"=dword:0x12
"MaxConnections"=dword:0x4e20
"MaxPendingDeliveries"=dword:0x7d0
"DB_Protocol"="ODBC"
"TxnMonitor"="COM"
"DbServer"="tpcc"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"=""
"COM_SinglePool"="YES"
```

Microsoft Internet Information Service Registry Parameters

Microsoft Internet Information Service Registry Parameters

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:0x19
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,00
"PoolThreadLimit"=dword:0xbe
"ThreadTimeout"=dword:0x15180

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
```

```
"Library"="infectrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:0x842
"Last Help"=dword:0x843
"First Counter"=dword:0x802
```

World Wide Web Service Registry Parameters

World Wide Web Service Registry Parameters

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC]
"Type"=dword:0x20
"Start"=dword:0x2
"ErrorControl"=dword:0x1
"ImagePath"=hex(2):43,00,3a,00,5c,00,57,00,49,00,4e,00,4e,00,54,00,5c,00,53,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00,6e,00,65,00,74,00,73,00,72,00,76,00,5c,00,69,00,6e,00,65,00,74,00,69,00,6e,00,66,00,6f,00,2e,00,65,00,78,00,65,00,00,00
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):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 snap-in."
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP]
"NOTE"="This is for backward compatibility only."
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP\Parameters]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters]
"MajorVersion"=dword:00000005
"MinorVersion"=dword:00000000
"InstallPath"="C:\WINNT\System32\inetsrv"
"CertMapList"="C:\WINNT\System32\inetsrv\iisrmap.dll"
"AccessDeniedMessage"="Error: Access is Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\WINNT\System32\LogFiles"
"AcceptExOutstanding"=dword:00000028
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots]
"/"="c:\inetpub\wwwroot,205"
"/Scripts"="c:\inetpub\scripts,1"
```

```

"/IISAdmin"="C:\\WINNT\\System32\\inetrv\\iisadmin,,1"
"/IISSamples"="c:\\inetpub\\iissamples,,1"
"/MSADC"="c:\\program files\\common files\\system\\msadc,,1"
"/IISHelp"="c:\\winnt\\help\\iishelp,,1"
"/_vti_bin"="C:\\Program Files\\Common Files\\Microsoft Shared\\Web Server
Extensions\\40\\isapi,,1"
"/Rpc"="C:\\WINNT\\System32\\RpcProxy,,1"
"/Printers"="C:\\WINNT\\web\\printers,,201"

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\P
erformance]
"Library"="w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"Last Counter"=dword:000008f2
"Last Help"=dword:000008f3
"First Counter"=dword:00000850
"First Help"=dword:00000851

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\S
ecurity]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,0
2,\\
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00,00,\\
00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\\
05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\\
20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\\
00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\\
00,05,20,00,00,00,23,02,00,00,72,00,73,00,01,01,00,00,00,00,00,05,12,00,00,\\
00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\E
num]
"0"="Root\\LEGACY_W3SVC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

```

RTE Input Parameters

Profile: 3636w_36rte
 File Path: C:\\benchcrf\\crusader_4way\\3636w_36rte.pro
 Version: 1.0.1

Number of Engines: 36

Name: DRIVER1
 Description: rte111
 Directory: c:\\benchcrf\\log\\driver1
 Machine: rtes2
 Parameter Set: ~Default
 Index: 0
 Seed: 25744
 Configured Users: 1010
 Pipe Name: DRIVER13910218
 Connect Rate: 400
 Start Rate: 0
 CLIENT_NURAND: 233
 CPU: 0

Name: DRIVER10
 Description: rte221
 Directory: c:\\benchcrf\\log\\driver10
 Machine: rtes2
 Parameter Set: ~Default
 Index: 900000000
 Seed: 25744

Configured Users: 1010
 Pipe Name: DRIVER105368953
 Connect Rate: 400
 Start Rate: 0
 CLIENT_NURAND: 233
 CPU: 1

Name: DRIVER11
 Description: rte122
 Directory: c:\\benchcrf\\log\\driver11
 Machine: rtes2
 Parameter Set: ~Default
 Index: 1000000000
 Seed: 25744
 Configured Users: 1010
 Pipe Name: DRIVER115414703
 Connect Rate: 400
 Start Rate: 0
 CLIENT_NURAND: 233
 CPU: 2

Name: DRIVER12
 Description: rte222
 Directory: c:\\benchcrf\\log\\driver12
 Machine: rtes2
 Parameter Set: ~Default
 Index: 1100000000
 Seed: 25744
 Configured Users: 1010
 Pipe Name: DRIVER125445828
 Connect Rate: 400
 Start Rate: 0
 CLIENT_NURAND: 233
 CPU: 3

Name: DRIVER13
 Description: rte123
 Directory: c:\\benchcrf\\log\\driver13
 Machine: rtes2
 Parameter Set: ~Default
 Index: 1200000000
 Seed: 25744
 Configured Users: 1010
 Pipe Name: DRIVER135478015
 Connect Rate: 400
 Start Rate: 0
 CLIENT_NURAND: 233
 CPU: 0

Name: DRIVER14
 Description: rte223
 Directory: c:\\benchcrf\\log\\driver14
 Machine: rtes2
 Parameter Set: ~Default
 Index: 1300000000
 Seed: 25744
 Configured Users: 1010
 Pipe Name: DRIVER145517468
 Connect Rate: 400
 Start Rate: 0
 CLIENT_NURAND: 233
 CPU: 1

Name: DRIVER15
 Description: rte124
 Directory: c:\\benchcrf\\log\\driver15
 Machine: rtes2
 Parameter Set: ~Default

Index: 1400000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER155550687
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: DRIVER16
Description: rte224
Directory: c:\benchrf\log\driver16
Machine: rtes2
Parameter Set: ~Default
Index: 1500000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER165594984
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 3

Name: DRIVER17
Description: rte134
Directory: c:\benchrf\log\driver17
Machine: rtes2
Parameter Set: ~Default
Index: 1600000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER178253984
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER18
Description: rte234
Directory: c:\benchrf\log\driver18
Machine: rtes2
Parameter Set: ~Default
Index: 1700000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER188295625
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER19
Description: rte132
Directory: c:\benchrf\log\driver19
Machine: rtes2
Parameter Set: ~Default
Index: 1800000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER198333625
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: DRIVER2
Description: rte211
Directory: c:\benchrf\log\driver2

Machine: rtes2
Parameter Set: ~Default
Index: 1000000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER24802593
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER20
Description: rte232
Directory: c:\benchrf\log\driver20
Machine: rtes2
Parameter Set: ~Default
Index: 1900000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER20122968234
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 3

Name: DRIVER21
Description: rte133
Directory: c:\benchrf\log\driver21
Machine: rtes2
Parameter Set: ~Default
Index: 2000000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER21226614218
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER22
Description: rte233
Directory: c:\benchrf\log\driver22
Machine: rtes2
Parameter Set: ~Default
Index: 2100000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER22226667312
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER23
Description: rte141
Directory: c:\benchrf\log\driver23
Machine: frte90
Parameter Set: ~Default
Index: 2050000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER23226716687
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER24

Description: rte241
Directory: c:\benchcrf\log\driver24
Machine: frte90
Parameter Set: ~Default
Index: 1950000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER24226857046
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER25
Description: rte142
Directory: c:\benchcrf\log\driver25
Machine: frte90
Parameter Set: ~Default
Index: 1850000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER25226902468
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER26
Description: rte242
Directory: c:\benchcrf\log\driver26
Machine: frte90
Parameter Set: ~Default
Index: 1750000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER26226980859
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER27
Description: rte143
Directory: c:\benchcrf\log\driver27
Machine: frte90
Parameter Set: ~Default
Index: 1650000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER27227025375
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER28
Description: rte243
Directory: c:\benchcrf\log\driver28
Machine: frte90
Parameter Set: ~Default
Index: 1550000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER28227063984
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER29
Description: rte144
Directory: c:\benchcrf\log\driver29
Machine: frte90
Parameter Set: ~Default
Index: 1450000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER29227139093
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER3
Description: rte112
Directory: c:\benchcrf\log\driver3
Machine: rtes2
Parameter Set: ~Default
Index: 200000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER34857234
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: DRIVER30
Description: rte244
Directory: c:\benchcrf\log\driver30
Machine: frte90
Parameter Set: ~Default
Index: 1350000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER30227178421
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER31
Description: rte115
Directory: c:\benchcrf\log\driver31
Machine: frte90
Parameter Set: ~Default
Index: 1250000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER3162687015
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER32
Description: rte125
Directory: c:\benchcrf\log\driver32
Machine: frte90
Parameter Set: ~Default
Index: 1150000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER3262743437
Connect Rate: 400
Start Rate: 0

CLIENT_NURAND: 233
CPU: 1

Name: DRIVER33
Description: rte135
Directory: c:\benchcrf\log\driver33
Machine: frte90
Parameter Set: ~Default
Index: 1050000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER3362792125
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER34
Description: rte235
Directory: c:\benchcrf\log\driver34
Machine: frte90
Parameter Set: ~Default
Index: 950000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER3462856218
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER35
Description: rte215
Directory: c:\benchcrf\log\driver35
Machine: frte90
Parameter Set: ~Default
Index: 850000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER3562898265
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER36
Description: rte225
Directory: c:\benchcrf\log\driver36
Machine: frte90
Parameter Set: ~Default
Index: 750000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER3662943343
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER4
Description: rte212
Directory: c:\benchcrf\log\driver4
Machine: rtes2
Parameter Set: ~Default
Index: 300000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER45155671

Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 3

Name: DRIVER5
Description: rte113
Directory: c:\benchcrf\log\driver5
Machine: rtes2
Parameter Set: ~Default
Index: 400000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER55193562
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER6
Description: rte213
Directory: c:\benchcrf\log\driver6
Machine: rtes2
Parameter Set: ~Default
Index: 500000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER65224953
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER7
Description: rte114
Directory: c:\benchcrf\log\driver7
Machine: rtes2
Parameter Set: ~Default
Index: 600000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER75257406
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: DRIVER8
Description: rte214
Directory: c:\benchcrf\log\driver8
Machine: rtes2
Parameter Set: ~Default
Index: 700000000
Seed: 25744
Configured Users: 1010
Pipe Name: DRIVER85300468
Connect Rate: 400
Start Rate: 0
CLIENT_NURAND: 233
CPU: 3

Name: DRIVER9
Description: rte121
Directory: c:\benchcrf\log\driver9
Machine: rtes2
Parameter Set: ~Default
Index: 800000000
Seed: 25744

Configured Users: 1010 Pipe Name: DRIVER95332359 Connect Rate: 400 Start Rate: 0 CLIENT_NURAND: 233 CPU: 0 Number of User groups: 36 Driver Engine: DRIVER1 IIS Server: client11 SQL Server: tpcc User: sa Protocol: Html w_id Range: 1 - 101 w_id Max Warehouse: 3636 Scale: Normal User Count: 1010 District id: 1 Scale Down: No Driver Engine: DRIVER18 IIS Server: client23 SQL Server: tpcc User: sa Protocol: Html w_id Range: 1718 - 1818 w_id Max Warehouse: 3636 Scale: Normal User Count: 1010 District id: 1 Scale Down: No Driver Engine: DRIVER10 IIS Server: client22 SQL Server: tpcc User: sa Protocol: Html w_id Range: 910 - 1010 w_id Max Warehouse: 3636 Scale: Normal User Count: 1010 District id: 1 Scale Down: No Driver Engine: DRIVER31 IIS Server: client11 SQL Server: tpcc User: sa Protocol: Html w_id Range: 3031 - 3131 w_id Max Warehouse: 3636 Scale: Normal User Count: 1010 District id: 1 Scale Down: No Driver Engine: DRIVER9 IIS Server: client12 SQL Server: tpcc User: sa Protocol: Html w_id Range: 809 - 909 w_id Max Warehouse: 3636 Scale: Normal User Count: 1010 District id: 1 Scale Down: No	Driver Engine: DRIVER20 IIS Server: client23 SQL Server: tpcc User: sa Protocol: Html w_id Range: 1920 - 2020 w_id Max Warehouse: 3636 Scale: Normal User Count: 1010 District id: 1 Scale Down: No Driver Engine: DRIVER14 IIS Server: client22 SQL Server: tpcc User: sa Protocol: Html w_id Range: 1314 - 1414 w_id Max Warehouse: 3636 Scale: Normal User Count: 1010 District id: 1 Scale Down: No Driver Engine: DRIVER28 IIS Server: client24 SQL Server: tpcc User: sa Protocol: Html w_id Range: 2728 - 2828 w_id Max Warehouse: 3636 Scale: Normal User Count: 1010 District id: 1 Scale Down: No Driver Engine: DRIVER11 IIS Server: client12 SQL Server: tpcc User: sa Protocol: Html w_id Range: 1011 - 1111 w_id Max Warehouse: 3636 Scale: Normal User Count: 1010 District id: 1 Scale Down: No Driver Engine: DRIVER12 IIS Server: client22 SQL Server: tpcc User: sa Protocol: Html w_id Range: 1112 - 1212 w_id Max Warehouse: 3636 Scale: Normal User Count: 1010 District id: 1 Scale Down: No Driver Engine: DRIVER22 IIS Server: client23 SQL Server: tpcc User: sa Protocol: Html w_id Range: 2122 - 2222 w_id Max Warehouse: 3636
---	--

Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER13
IIS Server: client12
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 1213 - 1313
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER23
IIS Server: client14
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 2223 - 2323
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER26
IIS Server: client24
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 2526 - 2626
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER27
IIS Server: client14
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 2627 - 2727
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER32
IIS Server: client12
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 3132 - 3232
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER24
IIS Server: client24
SQL Server: tpcc

User: sa
Protocol: Html
w_id Range: 2324 - 2424
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER25
IIS Server: client14
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 2425 - 2525
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER29
IIS Server: client14
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 2829 - 2929
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER15
IIS Server: client12
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 1415 - 1515
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER16
IIS Server: client22
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 1516 - 1616
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER3
IIS Server: client11
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 203 - 303
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER5
IIS Server: client11
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 405 - 505
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER30
IIS Server: client24
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 2930 - 3030
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER2
IIS Server: client21
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 102 - 202
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER33
IIS Server: client13
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 3233 - 3333
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER34
IIS Server: client23
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 3334 - 3434
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER35
IIS Server: client21
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 3435 - 3535
w_id Max Warehouse: 3636

Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER8
IIS Server: client21
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 708 - 808
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER36
IIS Server: client22
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 3536 - 3636
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER7
IIS Server: client11
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 607 - 707
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER4
IIS Server: client21
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 304 - 404
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER21
IIS Server: client13
SQL Server: tpcc
User: sa
Protocol: Html
w_id Range: 2021 - 2121
w_id Max Warehouse: 3636
Scale: Normal
User Count: 1010
District id: 1
Scale Down: No

Driver Engine: DRIVER19
IIS Server: client13
SQL Server: tpcc

User: sa
 Protocol: Html
 w_id Range: 1819 - 1919
 w_id Max Warehouse: 3636
 Scale: Normal
 User Count: 1010
 District id: 1
 Scale Down: No

Driver Engine: DRIVER6
 IIS Server: client21
 SQL Server: tpcc
 User: sa
 Protocol: Html
 w_id Range: 506 - 606
 w_id Max Warehouse: 3636
 Scale: Normal
 User Count: 1010
 District id: 1
 Scale Down: No

Driver Engine: DRIVER17
 IIS Server: client13
 SQL Server: tpcc
 User: sa
 Protocol: Html
 w_id Range: 1617 - 1717
 w_id Max Warehouse: 3636
 Scale: Normal
 User Count: 1010
 District id: 1
 Scale Down: No

Number of Parameter Sets: 1

		~Default					
		Default Parameter Set					
		Txn	Think	Key	RT	RT	Menu
		Weight	Time	Time	Delay	Fence	Delay
5.00	0.10	New Order	44.88	12.05	18.01		0.10
		Payment	43.03	12.05	3.01		0.10
5.00	0.10	Delivery	4.03	5.05	2.01		0.10
5.00	0.10	Stock Level	4.03	5.05	2.01		0.10
20.00	0.10	Order Status	4.03	10.05	2.01		0.10
5.00	0.10						

Appendix D: 60-Day Space

TPC-C 60-Day Space Requirements						
Warehouses	3,636				tpmC	45,230.03
Table	Rows	Data KB	Index KB	Extra 5% KB	8HR Space	Total Space KB
Warehouse	3,636	392	16	20.40		428.40
District	36,360	4,040	16	202.80		4,258.80
Item	100,000	9,528	32	478.00		10,038.00
New-Order	32,724,000	517,376	1,200		290,880.00	809,456.00
History	109,080,000	6,060,008	88		1,206,153.24	7,266,249.24
Orders	109,080,000	3,343,456	1,520,368		968,056.79	5,831,880.79
Customer	109,080,000	79,330,912	4,730,496	4,203,070.40		88,264,478.40
Order-Line	1,090,800,831	68,175,056	144,304		13,597,741.26	81,917,101.26
Stock	363,600,000	116,352,000	217,496	5,828,474.80		122,397,970.80
Totals		273,792,768	6,614,016	10,032,246.40	16,062,831.29	306,501,861.69
Segment	LogDev Cnt.	Segment Size	Needed	Overhead		Not Needed
misc	1	102,400,000	95,839,412	958,394		5,602,193.38
big	3	245,760,000	210,662,449	2,106,624		32,990,926.31
master, msdb,model	1	13,312	13,312			
tpcc_root	1	8,192	8,192			
tempdb	1	8,704	8,704			
Totals		348,190,208.00	306,532,069.69	3,065,018.62		38,593,119.69
Dynamic Space	77,578,520.00	Sum of Data for Order, Order-Line and History				
Static Space	215,925,529.02	Data + Index + 5% Space + Overhead - Dynamic Space				
Free Space	16,090,039.29	Total Segment Size - Dynamic Space - Static Space - Not needed				
Daily Growth	15,440,610.72	(Dynamic Space/W * 62.5)* tpmC				
Daily Spread	(7,067,876.79)	Free Space - 1.5 * Daily Growth (Zero If Negative)				
60-Day Space (KB)	1,142,362,172.44	Static Space + 60 (Daily Growth + Daily Spread)				
60-Day Space (GB)	1,089.44	60-Day Space in GB (Excludes OS,Paging and RDBMS Logs)				
Available (GB)	2,641.08	Total storage configured and available for database, minus logs, in RAID-0 configuration.				
Log File Storage Requirement						
Log Size (MB)	100,000.00	Total Size of Log File				
% Log Used	30.2276	% of Log File Used During Entire Run				
Total N-O Txn	6,684,733.00	Total Count of New-Order Transactions during Entire Run				
Log / N-O Txn	4.63	KB of Log per New-Order Transaction				
8 Hour Log (GB)	95.87	8 Hours of Log in GB (Excluding Space for Redundancy)				
Log Configured (GB)	101.58					
Disk Capacity	MB	GB				
18.2GB 10K RPM	17,736	16.93				
Space Usage	GB Needed		Disks Priced	Disk Size	GB Priced	GB Usable
60-Day (RAID-0)	1,089.44		156	18.2GB	2,641.08	2,641.08
					Total DB	2,641.08
8hr Log (RAID-1)	95.87		12	18.2GB	203.16	101.58
					Total Log	101.58
OS, SQL Server	4.00		1	18.2GB	16.93	16.93
Total Space	1,189.31		169		2,861.17	2,759.59

Appendix E: Third-Party Quotations

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
<http://www.microsoft.com/>

Microsoft

July 16, 2002

IBM Corporation
Chris King
3039 Cornwallis Road
Research Triangle Park,
NC 27709

Chris:

Here is the information you requested regarding pricing for several Microsoft products to be used in conjunction with your TPC-C benchmark testing.

All pricing shown is in US Dollars (\$).

Part Number	Description	Unit Price	Quantity	Price
810-00846	SQL Server 2000 Enterprise Edition <i>Per processor licensing</i> <i>Discount schedule: Open Program Level C</i>	\$16,541	4	\$66,164
C11-00821	Windows 2000 Server <i>Server license only - No CALs</i> <i>Discount schedule: Open Program - No Level</i>	\$738	2	\$1,476
C10-00475	Windows 2000 Advanced Server <i>Server license only - No CALs</i> <i>Discount schedule: Open Program - No Level</i>	\$2,399	1	\$2,399
048-00317	Visual C++ Professional 6.0 Win32	\$ 549	1	\$ 549
	3-year maintenance for above software	\$1,950	1	\$5,850

All products are currently orderable through Microsoft's normal distribution channels.

This quote is valid for the next 90 days.

If we can be of any further assistance, please contact Jamie Reding at (425) 703-0510 or jamiere@microsoft.com.

Reference ID: PCchki0216075188

Please include this Reference ID in any correspondence regarding this price quote.

7/17/2002 5:14:55 PM :


**COMPUTER
GIANTS**
ComputerGiants.com inc.

168 Madison Ave, 5th Floor

New York, NY 10016

Fax: 212-447-4489**Phone:** 212-447-4487

QUOTATION

Bill To:

IBM Attn : CHRIS KING

.

.

DURHAM, NC 27709

(919) 543-8799 : 919-486-2327

Ship To:

IBM Attn : CHRIS KING

.

.

DURHAM, NC 27709

Quote Date & Time	Order Status	E-mail
7/17/2002 5:12:46 PM	Quote	KCHRIS@US.IBM.COM

Quote Number	Sales Rep	Shipping Type	Terms
109434	Justin	Ground	Net 30

Quantity	Item	Description	Unit Price	Total
9	08P3834	MYLEX EXTREMERAID 2000 2INT CHAN 4EXT UHD CH U160 SCSI 32MB	\$1,280.00	\$11,520.00
			Subtotal	\$11,520.00
			Taxes	\$0.00
			Shipping	\$20.00
			Grand Total	\$11,540.00

Thank you for choosing Computer Giants.

Payment terms are always in advance unless otherwise specified in an agreement between buyer and ComputerGiants.com. All unpaid invoices shall bear interest at 11/2 % of the outstanding balance per month, commencing on the payment due date. In the event that ComputerGiants.com must send a sales invoice to an attorney for collection, the buyer agrees to pay any and all costs associated with such collection, including, but not limited to, attorney's fee and cost incurred prior to during and after a trial, and including, but not limited to, collection, bankruptcy or other creditor's rights proceeding. All COD transactions are secure (Bank Check or Money Order) or with a Credit Card Authorization on a Company Check or Personal Check.

All international Orders are Prepayment Wire Transfer Company Check etc. No exceptions.

The Computer Industry has an inherent volatility in pricing. Prices are subject to change.



Show Order

Notebooks

Networking

Wireless
Adapters
Hubs
Switches
Network Kits
USB
PCMCIA
Broadband
Print Servers
Modems
Multimedia
Transceivers
Cables
Routers
Storage
MP3 Players
IEEE-1394
LCD Monitor



NETLUX SW-5, 5-Port
10/100Mbps Dual Speed
FAST Ethernet Switch
\$35.95



D-Link DSS-5 5-port Dual Speed Switch
\$34.89

D-Link DSS-8+ 8-port
10/100Mbps Dual Speed
Switch
\$46.89



D-Link DSS-16+
D-Link 16-port Dual Speed
Switch
\$107.95



D-Link DSS-16
D-Link Rack Mount 16-port Dual Speed Switch
\$309.00



D-Link DSS-24+
D-Link RACK-MOUNT
24-port Dual Speed Switch
\$207.89



DES-1009G
D-Link DES-1009G 8-port
10/100 + 1-port 1000Mbps
Gigabit over Copper Switch
\$198.95



DES-3326
D-Link DES-3326 Layer 3 Switch with Gigabit ports
\$2,399.00



D-Link DES-3225G
D-Link 24-port 10/100Mbps
Ethernet Switch with
optional 100Mbps or
1000Mbps Gigabit Fiber port
options
\$789.95



D-Link DES-3624I
D-Link Stackable 24-port
10/100Mbps Ethernet Switch
with Optional 100Mbps Fiber
and 1000Mbps Gigabit Fiber
ports (Master Unit)
\$789.00



D-Link DES-3624
D-Link 24-port 10/100Mbps Ethernet Switch- Slave unit
for use with DES-3624I
\$689.00



D-Link DGS-3204
D-Link 4-port 1000Mbps
Gigabit Ethernet Switch
\$1,298.95



D-Link DGS-3208F
D-Link 8-port 1000Mbps
Gigabit Ethernet Switch
\$2,712.95



D-Link DES-5664TX
D-Link 64-port Modular 10/100Mbps and 1000Mbps
Ethernet Switch
\$2,350.00



D-Link DES-5600
D-Link 64-port 10/100Mbps
and 1000Mbps Ethernet
Switch Base Unit w/ CPU
only
\$1,136.00