TPC Benchmark™ C
Full Disclosure Report
for
IBM @server xSeries 350
using
Microsoft SQL Server 2000 Enterprise Edition
and
Windows 2000 Advanced Server

TPC-C
Rev. 3.5 Upgrade to Rev. 5.0

Submitted for Review
April 9, 2001

First Edition - April 2001

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.

Trademarks

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

The following terms used in this publication are trademarks of other companies as follows: TPC Benchmark, trademark of Transaction Processing Performance Council; Intel, Pentium and Xeon are trademarks or 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 350 configured as a client/server system. This report documents the full disclosure information required by the TPC Benchmark™ C Standard Specification, Revision 3.5, including the methodology used to achieve the reported results. All testing fully complied with this revision level.

The software used on the xSeries 350 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.

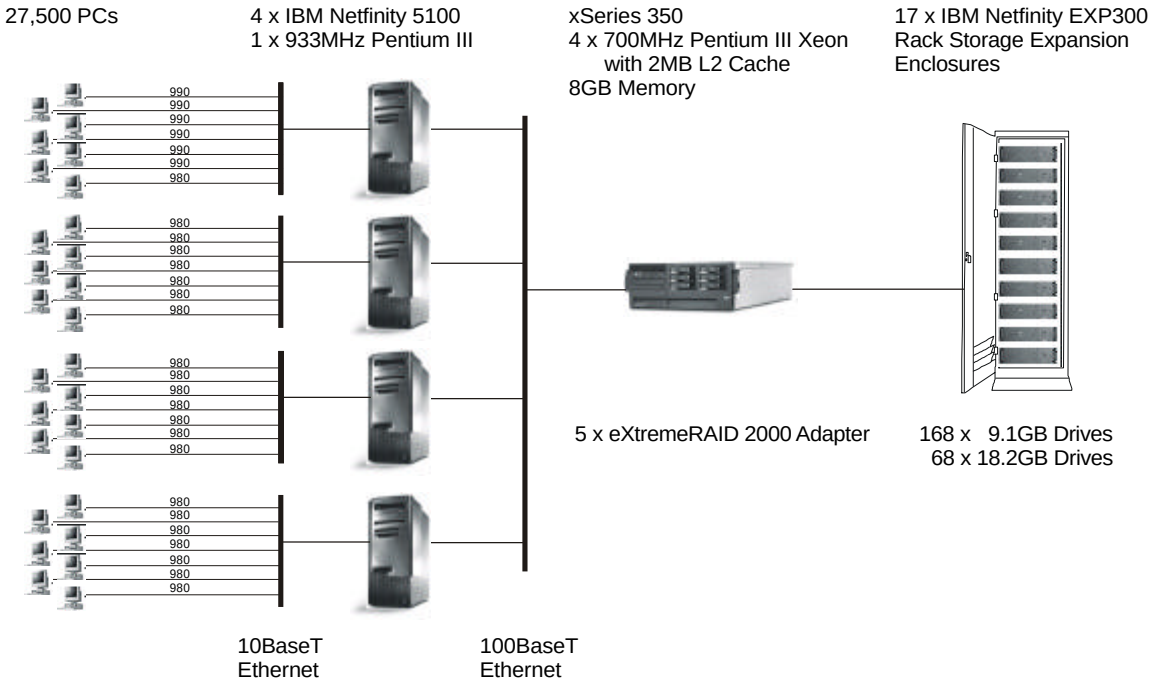
Hardware	Software	Total System Cost	tpmC	$/tpmC	Total Solution Availability Date
IBM @server xSeries 350	Microsoft SQL Server 2000 Enterprise Edition Microsoft Windows 2000 Advanced Server	$276,075	34,264.90	$8.06	Dec. 11, 2000

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

IBM	**IBM @server xSeries 350 c/s with Microsoft SQL Server 2000**	TPC-C Rev 5.0 Upgrade from Rev. 3.5 Report Date: April 9, 2001 (Original: December 12, 2000)

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$276,075	**34,264.90 tpmC**	**$8.06 /tpmC**	**Dec. 11, 2000**

Processors	Database Manager	Operating System	Other Software	Number of Users
4: Database 4: Client	**Microsoft® SQL Server 2000 Enterprise Edition**	**Microsoft Windows® 2000 Advanced Server**	**Microsoft Visual C++ 6.0 Win32 Microsoft COM+**	**27,500**



System Component	Qty	Server	Qty	Each of Four Clients
Processors	**4**	**700MHz Pentium III Xeon w/2MB L2 Cache**	**1**	**933MHz Pentium III w/256KB L2 Cache**
Memory	**16**	**512MB**	**1**	**128MB**
			1	**256MB**
Disk Controllers	**5**	**Mylex eXtremeRAID 2000**	**1**	**Ultra160 SCSI Onboard**
Disk Drives	**168**	**9.1GB (15000 rpm)**	**1**	**9.1GB Hard Disk**
	68	**18.2GB (15000 rpm)**		
	1	**9.1GB (10000rpm)**		
Total Storage		**2775.5GB**		
Tape Drive	**1**	**20/40GB SCSI Tape Drive**		

IBM Corporation	IBM @server xSeries 350 c/s with Microsoft SQL Server 2000	TPC-C Rev. 5.0 Upgrade from Rev. 3.5
		Report Date: April 9, 2001 (Original: December 12, 2000)

Description	Order Number	Third-Party Brand	Pricing	Unit Price	Qty	Ext. Price	3-Yr. Maint.*
Server Hardware							
xSeries 350 / 700MHz/2MB Pentium III Xeon'	86825RY		1	$5,369	1	$5,369	$2,495
700MHz/2MB L2 Cache Processor Upgrade	00N7944		1	2,879	3	8,637	0
512MB ECC SDRAM RDIMM Memory Kit	33L3117		1	749	15	11,235	0
Netinity 9.1GB 10K Ultra160 SCSI Drive	37L7204		1	275	1	275	0
Mylex eXtremeRAID 2000 Adapter**	E2000-4-32NB	Mylex	3	1,872	7	13,104	0
Netfinity 4.2M Ultra2 SCSI Cable	03K9311		1	105	18	1,890	0
10/100 Ethernet Server Adapter	06P3601		1	99	1	99	0
E54 14" (13.8" Viewable) Color Monitor*	6331N2N		1	159	1	159	90
20/40GB Internal SCSI Tape Drive	00N7991		1	769	1	769	0
Netfinity Rack*	9306900		1	1,725	2	3,450	600
Side Panel Kit	94G6669		1	195	1	195	0
Storage Hardware							
Netfinity EXP300 Rack Storage Enclosure*	35311RU		1	3,179	17	54,043	3,400
9.1GB 15K Ultra160 SCSI Drive	19K0655		1	405	168	68,040	0
18.2GB 15K Ultra160 SCSI Drive	19K0656		1	549	68	37,332	0
					Subtotal	$204,597	$6,585
Server Software							
Microsoft SQL Server 2000 Enterprise Editior	810-00846	Microsoft	2	16,541	4	$66,164	$0
Microsoft Windows 2000 Advanced Server	C10-00475	Microsoft	2	2,399	1	2,399	0
Three-Year Maintenance for Software		Microsoft	2	2,095	3		6,285
					Subtotal	$68,563	$6,285
Client Hardware							
Netfinity 5100 / 933MHz/256KB Pentium III*	865851Y		1	1,719	4	$6,876	$5,980
9.1GB 10K Ultra160 SCSI Drive	37L7204		1	275	4	1,100	0
256MB 133MHz ECC SDRAM RDIMM	33L3125		1	269	4	1,076	0
10/100 Ethernet Server Adapter	06P3601		1	99	4	396	0
E54 15" (13.8" Viewable) Color Monitor*	6331N2N		1	159	4	636	360
					Subtotal	$10,084	$6,340
Client Software							
Microsoft Windows 2000 Server with COM+	C11-00821	Microsoft	2	738	4	2,952	0
Microsoft Visual C++ Professional 6.0 Win32	048-00317	Microsoft	2	549	1	549	0
					Subtotal	$3,501	$0
User Connectivity							
8-Port 10/100Mbps Nway Fast Ethernet Switch***	NX-DSS8		4	25	7	175	0
					Subtotal	$175	$0
					Total	$286,920	$19,210
Large volume discount of 14% on IBM hardware; prices vary if purchased separately.			1	-14%		(30,055)	$0

Notes: * The standard 3-year warranties on IBM hardware has been upgraded to 7x24, 4-hour response time coverage. ** Five-year warranty. *** 10% or minimum 2 spares are added in place of on-site service (products have a 5-year return-to-vendor-warranty)
Pricing: 1 - IBM Corp.; 2 - Microsoft Corp.; 3 - Mylex ; 4 - Software House International.
Audited by Bradley J. Askins of InfoSizing, Inc.

Three-Year Cost of Ownership:	$276,075
tpmC Rating:	34,264.90
$ / tpmC:	$8.06

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

Numerical Quantities Summary

MQTh, Computed Maximum Qualified Throughput:	34,264.90 tpmC
% throughput difference, reported and reproducibility runs:	0.064%

Response Times (in seconds)	Average	Maximum	90 %-tile
New-Order	0.47	21.29	0.82
Payment	0.40	27.84	0.75
Delivery	0.13	7.21	0.16
Stock Level	1.20	11.88	1.75
Order Status	0.41	6.04	0.75
Delivery (Deferred)	0.21	0.89	0.35
Menu	0.13	13.46	0.17

Transaction Mix (in percent of total transactions)	Total Occurrences	Percent
New-Order	685,298	44.89
Payment	656,525	43.01
Delivery	61,504	4.03
Stock-Level	61,582	4.03
Order Status	61,643	4.04

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	18.03 / 120.51
Payment	3.02 / 12.04	3.00 / 0.00	3.03 / 120.50
Delivery	2.01 / 5.06	2.00 / 0.00	2.03 / 50.50
Stock Level	2.01 / 5.03	2.00 / 0.00	2.03 / 48.61
Order Status	2.02 / 10.06	2.00 / 0.00	2.03 / 100.50

Test Duration	
Ramp-up time	21 minutes
Measurement interval	20 minutes
Number of transactions (all types) completed in measurement interval	1,588,062
Ramp-down time	22 minutes
Number of checkpoints in measurement interval	1
Checkpoint interval	20 minutes

Table of Contents

Preface

The TPC Benchmark™ C was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry. This full disclosure report is based on the TPC Benchmark C Standard Specification Version 3.5, released October 25, 1999.

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

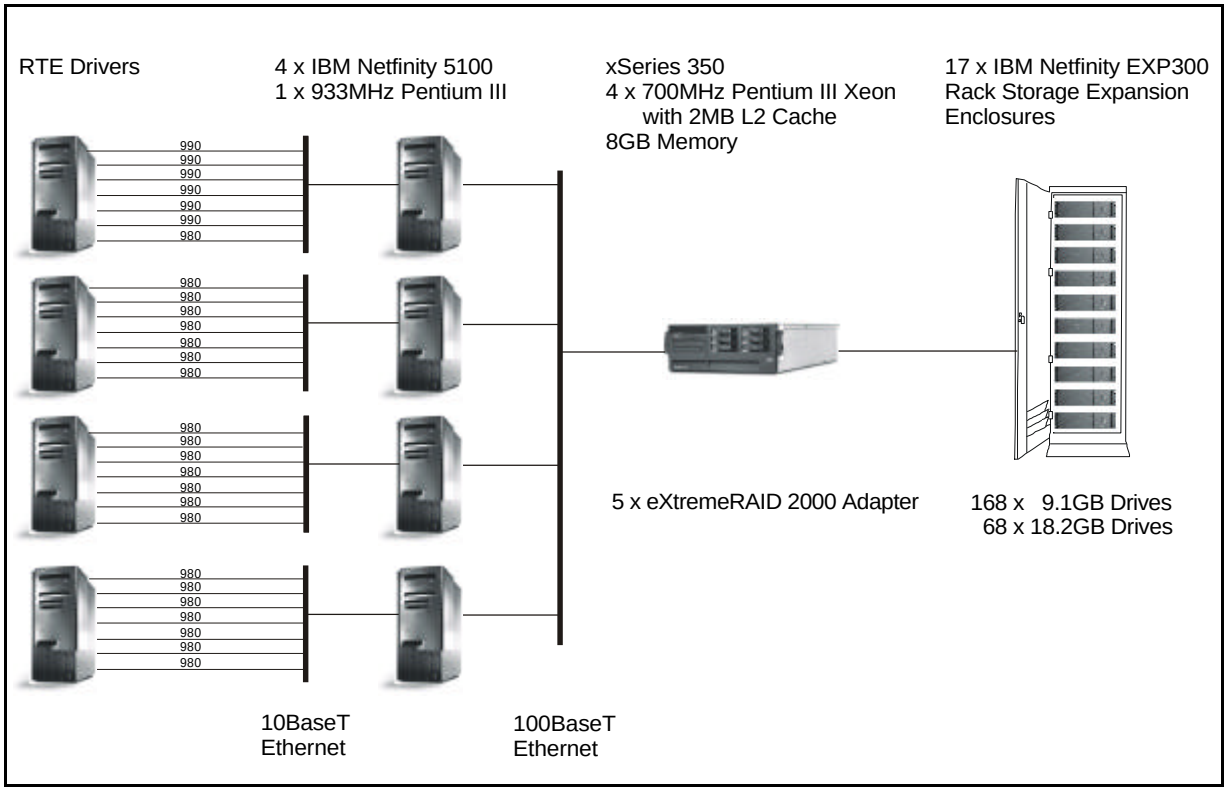
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences.

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

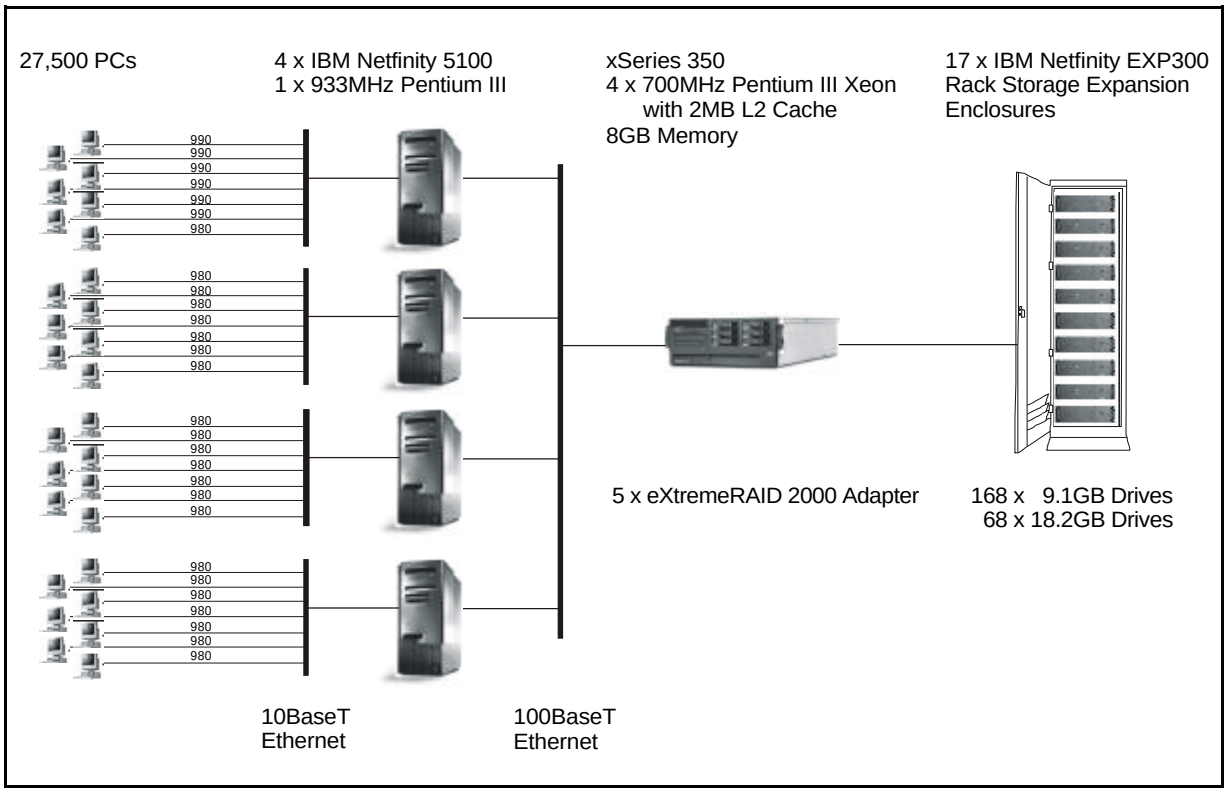
The Remote Terminal Emulator (RTE) used for these TPC Benchmark C tests is the Microsoft BenchCraft RTE. Under Version 3.5, the components of the configuration being emulated by the RTE are the workstations and the Ethernet hubs. Appendix C contains a listing of the RTE scripts and inputs used in the benchmark testing.

The benchmarked configuration used IBM Netfinity 5100 systems as 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 Configuration



Priced Configuration



Clause 1: Logical Database Design Related Items

Table Definitions

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

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

Physical Organization of the Database

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

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. (8.1.2.3)

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. (8.1.2.4)

Partitioning was not used in this benchmark.

Replication

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

Replication was not used in this benchmark.

Table Attributes

Additional and/or duplicated attributes in any table must be disclosed, along with a statement on the impact on performance (see Clause 1.4.7). (8.1.2.6)

No additional attributes were used in this benchmark.

Clause 2: Transaction and Terminal Profiles Related Items

Random Number Generation

The method of verification for the random number generation must be 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 IBM Netfinity 5100 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 IBM xSeries 350.

Transaction Profiles

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

The percentage of Delivery transactions that were skipped as a result of an insufficient number of rows in the NEW-ORDER table must be disclosed. (8.1.3.10)

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

Table 2-1. Transaction Statistics

New Order	Value (%)
Home warehouse order lines	99.00
Remote warehouse order lines	1.00
Rolled back transactions	1.01
Average number of items per order	10.00
Payment	
Home warehouse payment transactions	85.04
Remote warehouse payment transactions	14.96
Non-Primary Key Access	
Payment transactions using C_LAST	60.03
Order-Status transactions using C_LAST	60.26
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.89
Payment	43.01
Delivery	4.03
Stock Level	4.03
Order Status	4.04

Deferred Delivery Mechanism

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

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. (8.1.4.1)

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 150 users 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 gving SUM1.
2. A test was started under full load with all users submitting transactions.
3. One disk from the log array was removed. Since the disk was RAID-1 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. (8.1.5.1)

The database was originally built with 2,750 warehouses, and the audited run used all 2,750 warehouses.

Table 4-1. Initial Cardinality of Tables

Table Name	Rows
Warehouse	2,750
District	27,500
Item	100,000
New Order	24,750,000
History	82,500,000
Orders	82,500,000
Customer	82,500,000
Order Line	824,994,449
Stock	275,000,000
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. (8.1.5.2)

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	56 - 18.2GB	E: F:	39320MB 18250MB	Customer and Stock Misc.
2	56 - 9.1GB	G: H:	39320MB 18250MB	Customer and Stock Misc.
3	56 - 9.1GB	I: J:, Y:	39320MB 18250MB, 120000MB (NTFS)	Customer and Stock Misc., backup1
4	56 - 9.1GB	K: L:, Z:	39320MB 18250MB, 120000MB (NTFS)	Customer and Stock Misc., backup2
5	12 - 18.2GB	O:	86000MB	Log file

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/1, COBOL, read/write) used to implement the TPC-C transactions. If more than one interface/access language is used to implement TPC-C, each interface/access language must be described and a list of which interface/access language is used with which transaction type must be disclosed.*

Microsoft SQL Server 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). (8.1.5.5)

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. (8.1.6.1)

Measured tpmC: 34,264.90 tpmC

Price per tpmC: $8.06 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. (8.1.6.2)

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	0.47	21.29	0.82
Payment	0.40	27.84	0.75
Delivery	0.13	7.21	0.16
Stock Level	1.20	11.88	1.75
Order Status	0.41	6.04	0.75
Delivery (Deferred)	0.21	0.89	0.35
Menu	0.13	13.46	0.17

Keying/Think Times

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

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	18.03 / 120.51
Payment	3.02 / 12.04	3.00 / 0.00	3.03 / 120.50
Delivery	2.01 / 5.06	2.00 / 0.00	2.03 / 50.50
Stock Level	2.01 / 5.03	2.00 / 0.00	2.03 / 48.61
Order Status	2.02 / 10.06	2.00 / 0.00	2.03 / 100.50

Response Time Frequency Distribution Curves

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

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

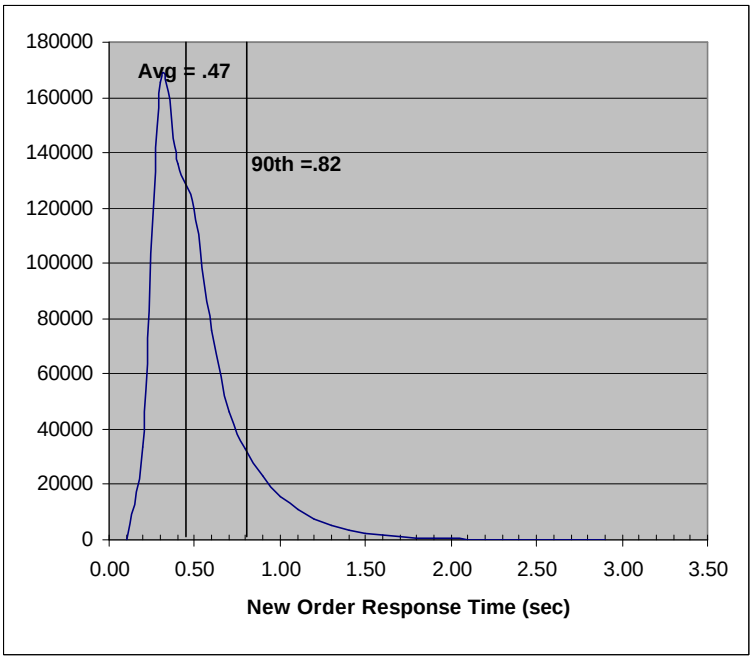


Figure 5-2. Payment Transaction - Response Time Frequency Distribution

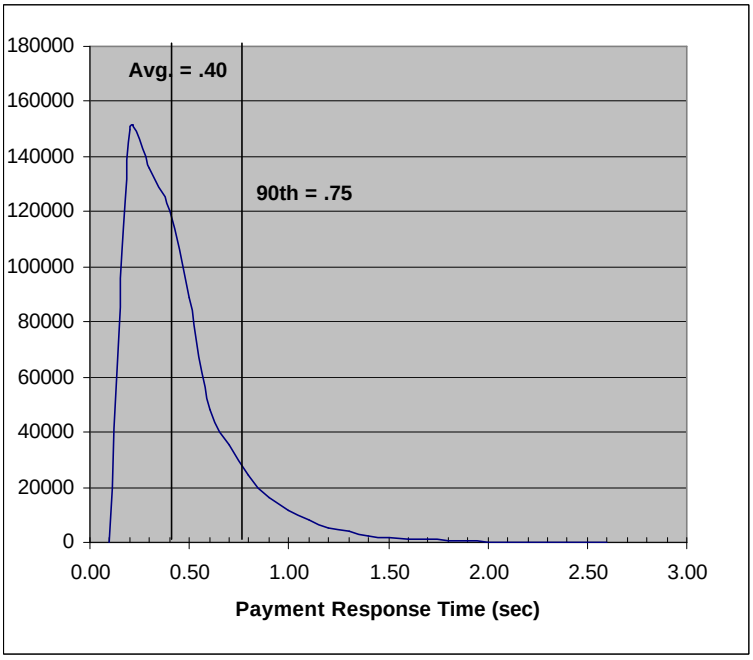


Figure 5-3. Order-Status Transaction - Response Time Frequency Distribution

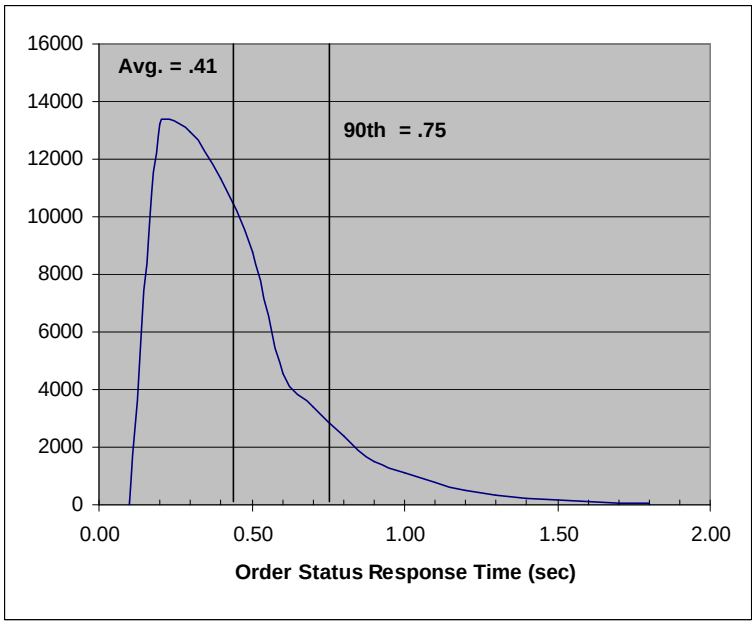


Figure 5-4. Delivery Transaction - Response Time Frequency Distribution

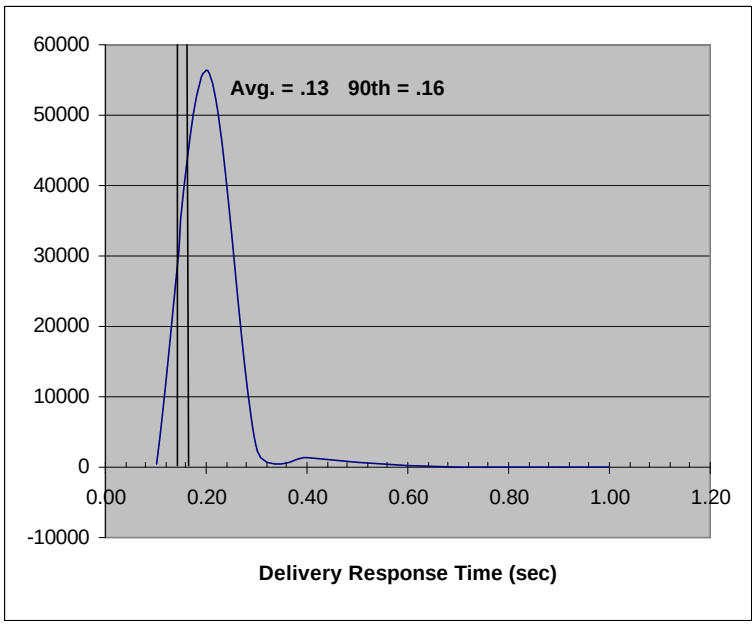
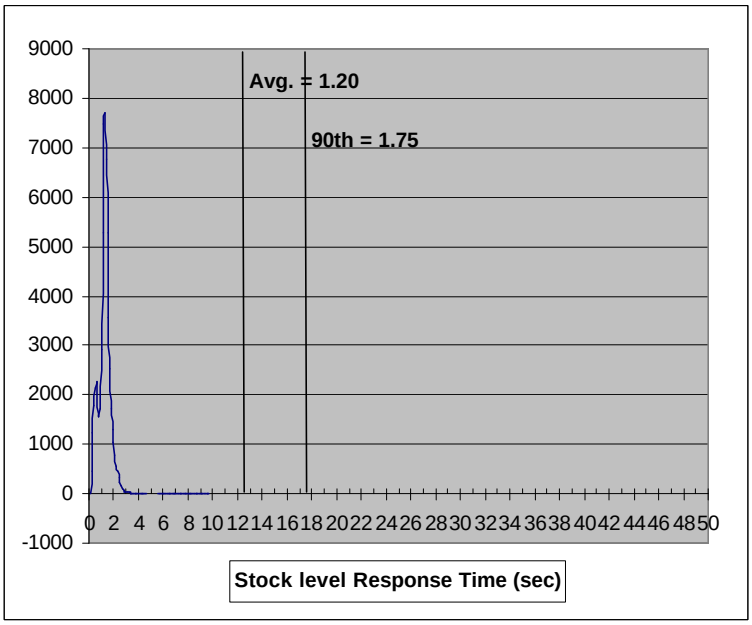


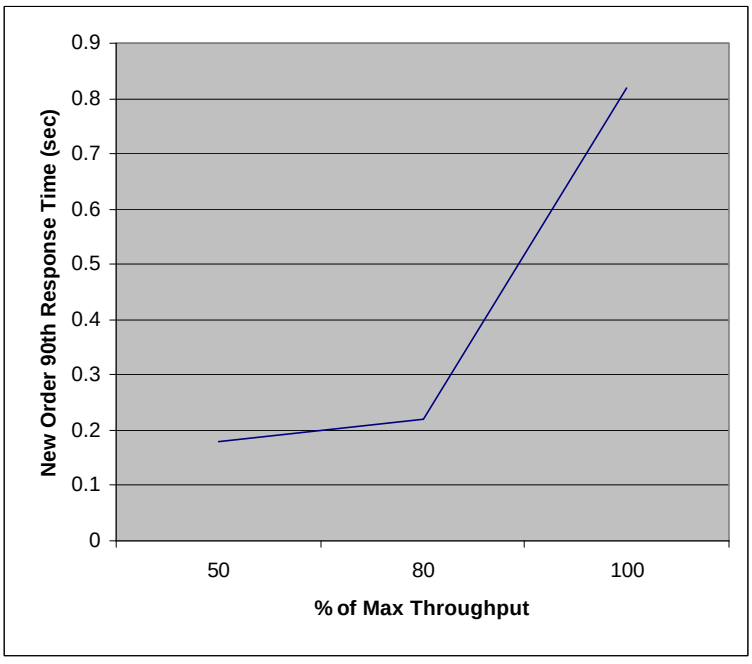
Figure 5-5. Stock-Level Transaction - Response Time Frequency Distribution



Performance Curve for Response Time vs. Throughput

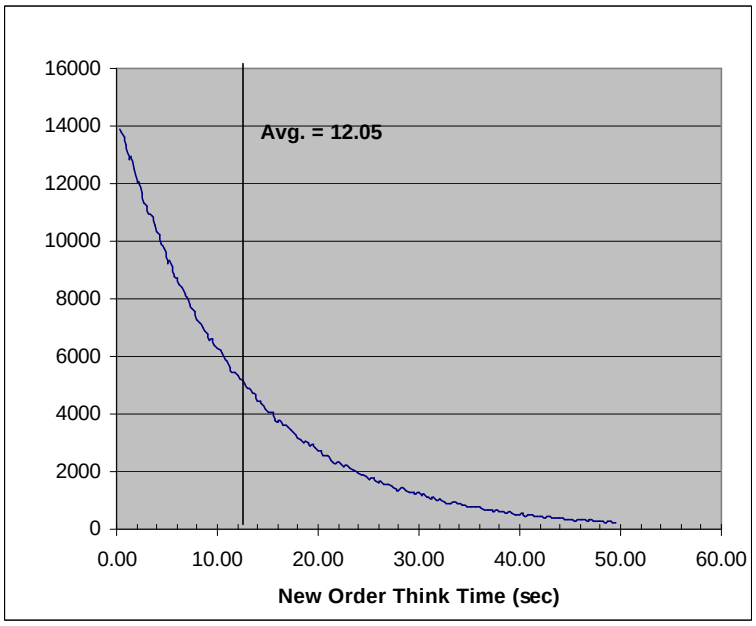
The performance curve for response time vs. throughput (see Clause 5.6.2) must be reported for the New-Order transaction. (8.1.6.5)

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



New Order Think Time Distribution

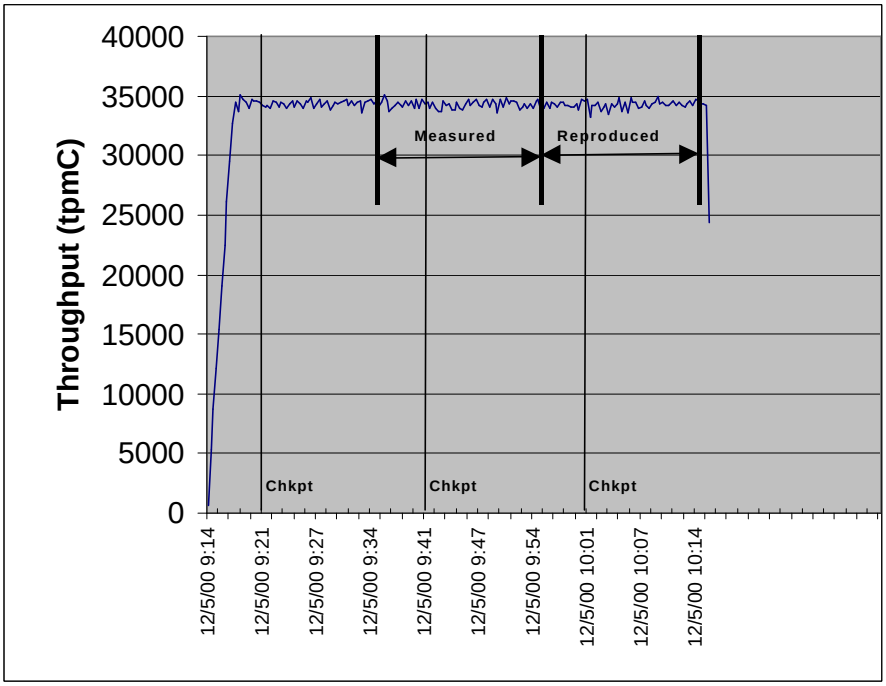
Figure 5-7. New-Order Think Time Distribution



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. (8.1.6.9)

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. (8.1.6.10)

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 20-minute intervals. Each measured run contained one checkpoint. 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.

Reproducibility Methodology

A description of the method used to determine the reproducibility of the measurement results must be reported. (8.1.6.11)

A repeatability measurement was taken on the IBM Netfinity 6000R for the same length of time as the measured run. The repeatability measurement was 34,242.90 tpmC.

Measurement Interval

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

The measurement interval was 20 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	1.01
Average number of items per order	10.00
Payment	
Home warehouse payment transactions	85.04
Remote warehouse payment transactions	14.96
Non-Primary Key Access	
Payment transactions using C_LAST	60.03
Order-Status transactions using C_LAST	60.26
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.89
Payment	43.01
Delivery	4.03
Stock Level	4.03
Order-Status	4.04

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 measured interval checkpoint started 6 minutes and 53 seconds after the start of the measurement interval. The checkpoint in the measured interval lasted 9 minutes.

The checkpoints were verified to be clear of the protected zones around the beginning and end of the measurement intervals. The checkpoint interval was 20 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. (8.1.7.1)

The RTE used was Microsoft BenchCraft 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. (8.1.7.2)

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). (8.1.7.4)

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. (8.1.7.5)

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 clients 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. (8.1.7.6)

The configuration did not require any operator intervention to sustain the reported throughput during the eight-hour period.

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. (8.1.8.1)

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. (8.1.8.2)

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. (8.1.8.3)

All hardware and 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. (8.1.8.4)

- Maximum Qualified Throughput: 34,264.90 tpmC
- Price per tpmC: $8.06 per tpmC
- Three-year cost of ownership: $276,075

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. (8.1.8.6)*

The component pricing based on usage is shown below:

- 4 Microsoft Windows 2000 Server
- 1 Microsoft Windows 2000 Advanced Server

- v 4 Microsoft SQL Server 2000 Enterprise Edition (based on per-processor price)
- v 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). (8.1.8.7) System pricing must include line item indication where non-sponsoring companies' brands are used. System pricing must also include line item indication of third-party pricing. (8.1.8.8)

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. (8.1.9.1)

This implementation of the TPC-C benchmark was audited by Bradley J. Askins 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 Benchmark™C," the Full Disclosure Report must have been submitted to the TPC Administrator as well as written permission obtained to distribute same. (8.2)

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

TPC
404 Balboa Street
San Francisco, CA 94118

INFO SIZING

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL CERTIFIED AUDITOR

Benchmark Sponsor: Richard Laviano
Mgr., xSeries Performance
IBM Server Group
3039 Cornwallis Road
Research Triangle Park, NC27709

August 31, 2001

I verified the TPC Benchmark™ C performance as well as the upgrade of the results from TPC-C Version 3.5 to TPC-C Version 5 for the following Client/Server configuration:

Platform:	**IBM @server xSeries 350 c/s**
Operating system:	**Microsoft Windows 2000 Advanced Server**
Database Manager:	**Microsoft SQL Server 2000 Enterprise Edition**
Transaction Manager:	**Microsoft COM+**

The results were:

CPU's Speed	Memory	Disks	NewOrder 90% Response Time	**tpmC**
Server: IBM @server xSeries 350				
4 x Pentium III Xeon (700 MHz)	8 GB Main (2MB L2 Cache per processor)	68 x 18.2 GB 169 x 9.1 GB	0.82 Seconds	**34264.90**
Four (4) Clients: Netfinity 5100 (Specification for each)				
1 x Pentium III (933 MHz)	384 MB Main (256KB L2 Cache per processor	1 x 9.1 GB	n/a	n/a

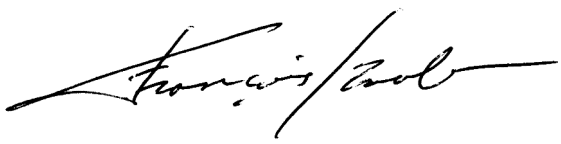
In my opinion, these performance results were produced in compliance with the TPC's requirements for the benchmark. The following verification items were given special attention:

- The database records were the proper size
- The database was properly scaled and populated

1373 North Franklin Street • Colorado Springs, CO 80903-2527 • Office: 719/473-7555 • Fax: 719/473-7554

- The required ACID properties were met
- The transactions were correctly implemented
- 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.
- All 90% response times were under the specified maximums
- At least 90% of all delivery transactions met the 80 Second completion time limit
- The reported measurement interval was 20 minutes (1200 seconds)
- The reported measurement interval was representative of steady state conditions
- One checkpoint was taken during the reported measurement interval
- The repeatability of the measured performance was verified
- The 60 day storage requirement was correctly computed
- The system pricing was verified for major components and maintenance

Respectfully Yours,



François Raab, President

Bradley J. Askins, Auditor

1373 North Franklin Street • Colorado Springs, CO 80903-2527 • Office: 719/473-7555 • Fax: 719/473-7554

Appendix A: Source Code

_delivery.h

```
/* Generated by IDL compiler version DEC DCE V2.0.0-6 */
#ifndef _delivery_v1_0_included
#define _delivery_v1_0_included
#ifndef IDLBASE_H
#include <dce\idlbase.h>
#endif
#include <dce\rpc.h>
#include "trpc/trpc.h"

#ifdef __cplusplus
    extern "C" {
#endif

#ifndef nbase_v0_0_included
#include "dce\nbase.h"
#endif
#ifndef trpcImports_v0_0_included
#include "trpc\trpcImports.h"
#endif
#ifndef mon_handle_v1_0_included
#include "tpm/mon\mon_handle.h"
#endif
#ifndef tpcc_types_v1_0_included
#include "tpcc_type.h"
#endif
#include <dce\rpcexc.h>
extern EXCEPTION encina_x_transaction_aborted;
extern EXCEPTION encina_x_server_shutdown;
extern EXCEPTION encina_x_permission_denied;
extern EXCEPTION encina_x_object_not_found;
extern EXCEPTION encina_x_empty_slot1;
extern EXCEPTION encina_x_empty_slot2;
extern EXCEPTION encina_x_empty_slot3;
extern EXCEPTION encina_x_empty_slot4;
extern EXCEPTION encina_x_empty_slot5;
extern EXCEPTION encina_x_undefined_exception;
extern void  IDL_STD_STDCALL _delivery_GetApplId(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
extern void  IDL_STD_STDCALL _impTPCCDelivery(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,
    /* [in, out] */ idl_char *dataP,
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
globalref mon_handle_t handle;
#if defined(__VMS) && (defined(__DECC) || defined(__cplusplus))
#pragma extern_model __save
#pragma extern_model __common_block __shr
#endif
typedef struct _delivery_v1_0_epv_t {
void ( IDL_STD_STDCALL *_delivery_GetApplId)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
void ( IDL_STD_STDCALL *_impTPCCDelivery)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,
    /* [in, out] */ idl_char *dataP,
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
} _delivery_v1_0_epv_t;
extern rpc_if_handle_t _delivery_v1_0_c_ifspec;
extern rpc_if_handle_t _delivery_v1_0_s_ifspec;
#if defined(__VMS) && (defined(__DECC) || defined(__cplusplus))
#pragma extern_model __restore
#endif

#ifdef __cplusplus
    }

#else
#endif
#endif
```

_neworder.h

```
/* Generated by IDL compiler version DEC DCE V2.0.0-6 */
#ifndef _neworder_v1_0_included
#define _neworder_v1_0_included
#ifndef IDLBASE_H
#include <dce\idlbase.h>
#endif
#include <dce\rpc.h>
#include "trpc/trpc.h"

#ifdef __cplusplus
    extern "C" {
#endif

#ifndef nbase_v0_0_included
#include "dce\nbase.h"
#endif
#ifndef trpcImports_v0_0_included
#include "trpc\trpcImports.h"
#endif
#ifndef mon_handle_v1_0_included
#include "tpm/mon\mon_handle.h"
#endif
#ifndef tpcc_types_v1_0_included
#include "tpcc_type.h"
#endif
#include <dce\rpcexc.h>
extern EXCEPTION encina_x_transaction_aborted;
extern EXCEPTION encina_x_server_shutdown;
extern EXCEPTION encina_x_permission_denied;
```

```
extern EXCEPTION encina_x_object_not_found;
extern EXCEPTION encina_x_empty_slot1;
extern EXCEPTION encina_x_empty_slot2;
extern EXCEPTION encina_x_empty_slot3;
extern EXCEPTION encina_x_empty_slot4;
extern EXCEPTION encina_x_empty_slot5;
extern EXCEPTION encina_x_undefined_exception;
extern void  IDL_STD_STDCALL _neworder_GetApplId(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
extern void  IDL_STD_STDCALL _impTPCCNewOrder(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,
    /* [in, out] */ idl_char *dataP,
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
extern void  IDL_STD_STDCALL _impTPCCNOInfo(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [out] */ dbInfo_data_t *dataP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
globalref mon_handle_t handle;
#if defined(__VMS) && (defined(__DECC) || defined(__cplusplus))
#pragma extern_model __save
#pragma extern_model __common_block __shr
#endif
typedef struct _neworder_v1_0_epv_t {
void ( IDL_STD_STDCALL *_neworder_GetApplId)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
void ( IDL_STD_STDCALL *_impTPCCNewOrder)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,
    /* [in, out] */ idl_char *dataP,
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
void ( IDL_STD_STDCALL *_impTPCCNOInfo)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [out] */ dbInfo_data_t *dataP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
} _neworder_v1_0_epv_t;
extern rpc_if_handle_t _neworder_v1_0_c_ifspec;
extern rpc_if_handle_t _neworder_v1_0_s_ifspec;
#if defined(__VMS) && (defined(__DECC) || defined(__cplusplus))
#pragma extern_model __restore
#endif

#ifdef __cplusplus
    }

#else
#endif
#endif
```

_orderstatus.h

```
/* Generated by IDL compiler version DEC DCE V2.0.0-6 */
#ifndef _orderstatus_v1_0_included
#define _orderstatus_v1_0_included
#ifndef IDLBASE_H
#include <dce\idlbase.h>
#endif
#include <dce\rpc.h>
#include "trpc/trpc.h"

#ifdef __cplusplus
    extern "C" {
#endif

#ifndef nbase_v0_0_included
#include "dce\nbase.h"
#endif
#ifndef trpcImports_v0_0_included
#include "trpc\trpcImports.h"
#endif
#ifndef mon_handle_v1_0_included
#include "tpm/mon\mon_handle.h"
#endif
#ifndef tpcc_types_v1_0_included
#include "tpcc_type.h"
#endif
#include <dce\rpcexc.h>
extern EXCEPTION encina_x_transaction_aborted;
extern EXCEPTION encina_x_server_shutdown;
extern EXCEPTION encina_x_permission_denied;
extern EXCEPTION encina_x_object_not_found;
extern EXCEPTION encina_x_empty_slot1;
extern EXCEPTION encina_x_empty_slot2;
extern EXCEPTION encina_x_empty_slot3;
extern EXCEPTION encina_x_empty_slot4;
extern EXCEPTION encina_x_empty_slot5;
extern EXCEPTION encina_x_undefined_exception;
extern void  IDL_STD_STDCALL _orderstatus_GetApplId(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
```

```
#endif
);
extern void  IDL_STD_STDCALL _impTPCCOrderStatus(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,
    /* [in, out] */ idl_char *dataP,
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
globalref mon_handle_t handle;
#if defined(__VMS) && (defined(__DECC) || defined(__cplusplus))
#pragma extern_model __save
#pragma extern_model __common_block __shr
#endif
typedef struct _orderstatus_v1_0_epv_t {
void ( IDL_STD_STDCALL *_orderstatus_GetApplId)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
void ( IDL_STD_STDCALL *_impTPCCOrderStatus)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,
    /* [in, out] */ idl_char *dataP,
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
} _orderstatus_v1_0_epv_t;
extern rpc_if_handle_t _orderstatus_v1_0_c_ifspec;
extern rpc_if_handle_t _orderstatus_v1_0_s_ifspec;
#if defined(__VMS) && (defined(__DECC) || defined(__cplusplus))
#pragma extern_model __restore
#endif

#ifdef __cplusplus
    }

#else
#endif
#endif
```

_payment.h

```
/* Generated by IDL compiler version DEC DCE V2.0.0-6 */
#ifndef _payment_v1_0_included
#define _payment_v1_0_included
#ifndef IDLBASE_H
#include <dce\idlbase.h>
#endif
#include <dce\rpc.h>
#include "trpc/trpc.h"

#ifdef __cplusplus
    extern "C" {
#endif

#ifndef nbase_v0_0_included
#include "dce\nbase.h"
#endif
#ifndef trpcImports_v0_0_included
#include "trpc\trpcImports.h"
#endif
#ifndef mon_handle_v1_0_included
#include "tpm/mon\mon_handle.h"
#endif
#ifndef tpcc_types_v1_0_included
#include "tpcc_type.h"
#endif
#include <dce\rpcexc.h>
extern EXCEPTION encina_x_transaction_aborted;
extern EXCEPTION encina_x_server_shutdown;
extern EXCEPTION encina_x_permission_denied;
extern EXCEPTION encina_x_object_not_found;
extern EXCEPTION encina_x_empty_slot1;
extern EXCEPTION encina_x_empty_slot2;
extern EXCEPTION encina_x_empty_slot3;
extern EXCEPTION encina_x_empty_slot4;
extern EXCEPTION encina_x_empty_slot5;
extern EXCEPTION encina_x_undefined_exception;
extern void  IDL_STD_STDCALL _payment_GetApplId(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
extern void  IDL_STD_STDCALL _impTPCCPayment(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,
    /* [in, out] */ idl_char *dataP,
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
globalref mon_handle_t handle;
#if defined(__VMS) && (defined(__DECC) || defined(__cplusplus))
#pragma extern_model __save
#pragma extern_model __common_block __shr
#endif
typedef struct _payment_v1_0_epv_t {
void ( IDL_STD_STDCALL *_payment_GetApplId)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
void ( IDL_STD_STDCALL *_impTPCCPayment)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,
    /* [in, out] */ idl_char *dataP,
```

```
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
} _payment_v1_0_epv_t;
extern rpc_if_handle_t _payment_v1_0_c_ifspec;
extern rpc_if_handle_t _payment_v1_0_s_ifspec;
#if defined(__VMS) && (defined(__DECC) || defined(__cplusplus))
#pragma extern_model __restore
#endif

#ifdef __cplusplus
    }

#else
#endif
#endif
```

_stocklevel.h

```
/* Generated by IDL compiler version DEC DCE V2.0.0-6 */
#ifndef _stocklevel_v1_0_included
#define _stocklevel_v1_0_included
#ifndef IDLBASE_H
#include <dce\idlbase.h>
#endif
#include <dce\rpc.h>
#include "trpc/trpc.h"

#ifdef __cplusplus
    extern "C" {
#endif

#ifndef nbase_v0_0_included
#include "dce\nbase.h"
#endif
#ifndef trpcImports_v0_0_included
#include "trpc\trpcImports.h"
#endif
#ifndef mon_handle_v1_0_included
#include "tpm/mon\mon_handle.h"
#endif
#ifndef tpcc_types_v1_0_included
#include "tpcc_type.h"
#endif
#include <dce\rpcexc.h>
extern EXCEPTION encina_x_transaction_aborted;
extern EXCEPTION encina_x_server_shutdown;
extern EXCEPTION encina_x_permission_denied;
extern EXCEPTION encina_x_object_not_found;
extern EXCEPTION encina_x_empty_slot1;
extern EXCEPTION encina_x_empty_slot2;
extern EXCEPTION encina_x_empty_slot3;
extern EXCEPTION encina_x_empty_slot4;
extern EXCEPTION encina_x_empty_slot5;
extern EXCEPTION encina_x_undefined_exception;
extern void  IDL_STD_STDCALL _stocklevel_GetApplId(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
extern void  IDL_STD_STDCALL _impTPCCStockLevel(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,
    /* [in, out] */ idl_char *dataP,
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
globalref mon_handle_t handle;
#if defined(__VMS) && (defined(__DECC) || defined(__cplusplus))
#pragma extern_model __save
#pragma extern_model __common_block __shr
#endif
typedef struct _stocklevel_v1_0_epv_t {
void ( IDL_STD_STDCALL *_stocklevel_GetApplId)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t handle,
    /* [out] */ trpc_byteData_t applString,
    /* [out] */ idl_ulong_int *applStringLength,
    /* [out] */ trpc_byteData_t address,
    /* [out] */ idl_ulong_int *addressLength,
    /* [out] */ error_status_t *c_status,
    /* [out] */ error_status_t *f_status
#endif
);
void ( IDL_STD_STDCALL *_impTPCCStockLevel)(
#ifdef IDL_PROTOTYPES
    /* [in] */ handle_t trpc_h,
    /* [in] */ idl_long_int length,
    /* [in, out] */ idl_char *dataP,
    /* [in, out] */ data_header *headerP,
    /* [in] */ trpc_byteData_t applAndAddress,
    /* [in] */ idl_ulong_int applAndAddressLength,
    /* [in] */ trpc_callbackData_t inCallbackData,
    /* [in] */ idl_ulong_int numOfInCallbackData
#endif
);
} _stocklevel_v1_0_epv_t;
extern rpc_if_handle_t _stocklevel_v1_0_c_ifspec;
extern rpc_if_handle_t _stocklevel_v1_0_s_ifspec;
#if defined(__VMS) && (defined(__DECC) || defined(__cplusplus))
#pragma extern_model __restore
#endif

#ifdef __cplusplus
    }

#else
#endif
#endif
```

clientutils.c

```
/* client_utils.c
*/

#include <stdio.h>
#include <time.h>
#include <windows.h>
#include <winperf.h>
#include <winsock.h>
#include "client_utils.h"
```

```
#define Li2Double(x) ((double)((x).HighPart) * 4.294967296E9 +
(double)((x).LowPart))

static LARGE_INTEGER pFreq;
static double sFreq;
static int print_thread_id = 1;
static int user_id = 0;
static char *user_code = "C";

/*
 *  get_thread_id
 *      A function that returns the thread ID of the current thread
 */
static int get_thread_id()
{
   return(GetCurrentThreadId());
}


/*
 *  get_prefix
 *              Format the output prefix for printing:
 *              It contains the user_id, 'C' or 'T' depending on whether it
 *              is a terminal or a client and optional a thread identifier
 *              The prefix is written in the buffer passed in by the caller.
 */
static void get_prefix(char *buffer)
{
   if (print_thread_id) {
                           int thread_id = get_thread_id();
                           sprintf(buffer,  "%s(%d-%s-%d)%s",
                           user_id < 10 ? "  " : user_id < 100 ? " " : "",
                           user_id,
                           user_code,
                           thread_id,
                           thread_id < 10 ? " " : "");
   } else {
                           sprintf(buffer,  "%s(%2d-%s)",
                           user_id < 10 ? " " : "", user_id, user_code);
   }
}

/*
 * err_printf
 *      A var-arg function that appends the current time and
 *      other data to the print request and sends it to stderr
 *      if it is not a web client, to a file if it is
 */
void err_printf(char *format, ...)
{
   time_t cur_timet;
   char time_str[30];
   char line_prefix[50];
   va_list ap;

   va_start(ap, format);

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

   get_prefix(line_prefix);

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

   va_end(ap);
}


/*
 *  encina_error_message
 *
 *  Report an encina error message by interpreting it and writing
 *  it to both the logfile (if any) and to standard error
 */
void encina_error_message(char *msg, unsigned long n)
{
   char errorMsg[ENCINA_MAX_STATUS_STRING_SIZE];
   encina_StatusToString(n, ENCINA_MAX_STATUS_STRING_SIZE,
errorMsg);
   err_printf("ERROR: %s.  Error code = %s (%d 0x%x) \n", msg, errorMsg, n, n);
}

int get_time_init()
{
    QueryPerformanceFrequency(&pFreq);
    sFreq=Li2Double(pFreq);
    return 0;
}

int get_local_time(time_type *timeP)
{
   double cur_t;
   LARGE_INTEGER counter;

   QueryPerformanceCounter(&counter);
   cur_t = Li2Double(counter) / sFreq;
   timeP->sec = (long)cur_t;
/*    timeP->usec = ((long)cur_t - timeP->sec) * 1000000;*/
   timeP->usec = (long)((cur_t - timeP->sec) * 1000000);
              return 0;
}

/*
 *  time_diff_ms
 *      Return the difference in miliseconds 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);
}
```

clientutils.h

```
#ifndef TPCC_CLIENT_UTILS_H
#define TPCC_CLIENT_UTILS_H

#include <stdio.h>
#include <time.h>
#include <dce/rpc.h>
#include <dce/dce_error.h>
#include <encina/encina.h>
#include <stdlib.h>
#include <utils/trace.h>
#include <winsock.h>
#include "mon_client.h"
#include "../include/tpcc_type.h"

extern FILE * errtpcc;
extern FILE *logtpcc;
extern int debug;
extern char log_file_name[];
extern void logprintf( char *format, ...);
extern void err_printf( char *format, ...);
extern void encina_error_message(char *msg, unsigned long n);
extern int time_diff_ms(struct timeval *t2, struct timeval *t1);

typedef struct {
   int num;
   int errs;
   double RTtotal[2]; // 1 for server RT and 0 for client RT
   int RTcount;
} tran_info_t;

/*
 * total_tran_count_t
 *
 * structure that holds the total count of transaction of each type
 * as well as the reposne times.
 *
 */
typedef struct {
   tran_info_t tran[MAX_TRAN_TYPE + 1];
   int errors;
   double time;
} total_tran_count_t;

/* enc_status_t
 * structure that holds error information
 */
typedef struct {
   int status;
   int line;
   char file[268];
   unsigned long encinaError;
   char errorMsg[ENCINA_MAX_STATUS_STRING_SIZE];
} enc_status_t;

#define FALSE 0
#define TRUE  1

#define DPRINT(args)  if (0) err_printf args


#define CHECK_ENVIRON(str,var)  if (str == NULL) { fprintf(ERROUT, \
                                "%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  */
```

databuf.h

```
/*
 *     databuf.h
 *
 * $Revision: 1.1 $
 * $Date: 1998/11/06 21:10:11 $
 * $Log:    databuf.h,v $
 * Revision 4.2  95/05/16  10:55:31  10:55:31  tpcc (TPCC Benchmark)
 * Added necessary RCS ident strings
 *
 * Revision 4.1  95/05/09  15:21:02  15:21:02  strue (Scott Truesdale)
 * New code from Transarc - initial version
 *
 * Revision 3.2  95/04/03  17:43:09  17:43:09  strue (Scott Truesdale)
 * Changes from Transarc - added sql error handling in client; cleaned up debug
handling with macros; added check on db paramters via call to server.
 *
 * Revision 3.1  95/04/03  15:10:30  15:10:30  strue (Scott Truesdale)
 * Base of rev 3 - shipped to transarc
 *
 *
 *
 *
 *  $TALog: databuf.h,v $
 * Revision 1.1  1998/11/06  21:10:11  dongfeng
 * - Move all files common to client and server to tpcc/common
 *   directory
 * [added by delta dongfeng-23677-TPCC-new-directory-structures, r1.1]
 *
 * Revision 1.3  1998/10/22  15:33:04  wenjian
 * Make changes to Encina server code to connect with SQL server and add
 * callsql.c and sql directory.
 *
 * Add ERR_BAD_ITEM_ID, which is returned by SLQnew and same as
INVALID_NEWO
 * [from r1.2 by delta wenjian-23529-TPCC-integrate-with-SQL-server, r1.1]
 *
 * Revision 1.2  1998/01/23  15:07:47  oz
 * - Updated the SP TPCC directory to the latest files used
 *   during the SP tpcc audit.
 * [from r1.1 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
 *
 * Revision 1.1  1997/04/20  11:57:57  oz
 * - This is the code base modified at IBM Poughkeepsie
 *   by Ofer Zajicek and Radha Sivaramakrishnan for the
 *   SP scaling test for TPCC.
 * [added by delta oz-19782-TPCC-add-ibm-sp-code, r1.1]
 *
 * Revision 1.31  1995/10/30  19:10:54  oz
 * [merge of changes from 1.29 to 1.30 into 1.27]
 *
 * Revision 1.30  1995/10/27  15:41:30  oz
 * - Modified the tpc-c code to work with the new informix
 *   sql code that is in ex_trans.ec
 * [from r1.29 by delta oz-16761-TPCC-modify-code-to-work-with-oracle, r1.1]
 *
 * Revision 1.27  1995/10/20  18:44:30  ctipper
 * [merge of changes from 1.17 to 1.25 into 1.22]
 *
 * Revision 1.25  1995/10/20  18:15:34  ctipper
 * Incorporate changes per code review.
 *
 * - add DISTRIBUTED_TRAN_FAILED, TPCC_DB_INFO_PARTIAL, and
 * TPCC_DB_INFO_FAILED error codes to tpcc_rc_t
 * - got rid of MAX_NUM_SERVERS variables
 * [from r1.23 by delta ctipper-16547-TPCC-more-distributed-trans, r1.2]
 *
 * Revision 1.23  1995/10/13  17:00:26  ctipper
 * This delta encompasses all changes necessary to do distributed, XA
 * transactions with the TPCC benchmark. This includes the changes
 * necessary to build with Informix version 6.
 *
 * Each client still talks to only one server, however, if a distributed
 * transaction is necessary, the client sends the request to a different
 * interface of that server which then forwards all or part of the
 * request on to the appropriate remote server.
 *
 * - added new error codes to the tpcc_rc_t enumeration.
 * - defined MAX_NUM_SERVERS to be 10
 * [from r1.19 by delta ctipper-16547-TPCC-more-distributed-trans, r1.1]
 *
 * Revision 1.19  1995/09/20  21:02:39  oz
 * -Corrected code for the payment transaction
 * - The distributed case now no longer uses
 *   stored procedures
 * [from r1.18 by delta oz-16547-TPCC-add-distributed-transactions, r1.2]
 *
 * Revision 1.18  1995/09/20  17:51:10  oz
 * - Added distributed transactions for the new order and
 *   payment transaction
 *
 * - Added new error codes
 * [from r1.17 by delta oz-16547-TPCC-add-distributed-transactions, r1.1]
 *
 * Revision 1.22  1995/10/02  20:31:07  oz
 * - Corrected definition of ERROR()
 * [from r1.21 by delta oz-16638-tpcc-modify-terminal-for-RTE, r1.3]
 *
 * Revision 1.21  1995/10/02  18:51:45  oz
 * - Added definitions needed for utils.c and liberty.c
 * [from r1.20 by delta oz-16638-tpcc-modify-terminal-for-RTE, r1.2]
 *
 * Revision 1.20  1995/10/02  15:52:35  oz
 * - Modified the TPC-C benchmark to be compatible with the RTE.
 * - There are now 3 terminal processes:
 *     emulator: the old terminal process with a built in
 *                 simple emulator
 *     curses: An interactive terminal process using curses
 *     liberty: An interactive terminal process to be used with
 *                the RTE compatible with the liberty freedom terminal.
 *
 * - Define TRUE and FALSE only if they are not already defined.
 *   (curses.h defines TRUE)
 * - Removed READ_TO_DATE and YEAR_TO_SECOND
 * - Added term_type_t
 * - Added
 *            GOOD_INPUT (0)
 *            WRONG_INPUT (10)
 * [from r1.17 by delta oz-16638-tpcc-modify-terminal-for-RTE, r1.1]
 *
 * Revision 1.17  1995/07/28  15:28:23  oz
 * - Added a -null and -no_marshalling option to TPCC
 *
 * - Added INVALID_TRAN_TYPE return code
 * [from r1.16 by delta oz-16070-TPCC-add-null-and-marshalling-test, r1.1]
 *
 * Revision 1.16  1995/07/18  17:02:38  oz
 * - Added a DCE_ERROR error code
 * [from r1.15 by delta oz-15938-TPCC-add-dce-only-client, r1.1]
 *
 * Revision 1.15  1995/05/22  19:50:48  shl
 * [merge of changes from 1.12 to 1.13 into 1.14]
 *
 * Revision 1.13  1995/05/18  15:11:27  oz
```

```
 * [from r1.12 by delta oz-15290-TPCC-incorporate-hp-drop-of-05-16-95, r1.1]
 *
 * Revision 1.14  1995/05/22  17:26:35  ctipper
 * [merge of changes from 1.5 to 1.9 into 1.11]
 *
 * [*** log entries omitted ***]
 *
 */

#ifndef __TPCC_DATABUF_H__
#define __TPCC_DATABUF_H__

#define I_NAME_LEN      24
#define I_DATA          50
#define W_NAME_LEN      10
#define ADDR_LEN        20
#define STATE_LEN       2
#define ZIP_LEN         9
#define DIST_INFO_LEN   24
#define S_DATA_LEN      50
#define D_NAME_LEN      10
#define H_DATA_LEN      24
#define CARRIER_LEN     2
#define C_LAST_LEN              17
#define C_MID_LEN       2
#define PHONE_LEN       16
#define CREDIT_LEN      2
#define C_DATA_LEN      500
#define BC_DTA_LEN              23

#define YEAR_TO_DATE   1
#define YEAR_TO_SECOND 2

#define ERROR(x) fprintf(stderr,"Error: %s\n",#x),exit(11)

#define MAX_STR_LEN     255
#define MAX_OL                  15

#ifndef TRUE
#define TRUE    1
#endif
#ifndef FALSE
#define FALSE   0
#endif

#define CANCEL          -1

#define DATETIME_LEN    19

#define D_PER_W         10

#define COLLECTOR1                      /* ctipper 5/3/95 */

#define ERR_BAD_ITEM_ID  1  /* copied from sql/tpcc.h */
#define RPC_ERROR               -2
#define SUCCESS_CODE                    0

#define CHAR_NULL       '\0'            /* strue 1/23/95 */

typedef enum {
   liberty_term,
   curses_term,
   emulator_term
} term_type_t;

typedef enum {
   GOOD_INPUT = 0,

   SQL_ERROR = 2,
   DCE_ERROR = 4,
   NO_SUCH_LAST_NAME = 5,
   INVALID_TRAN_TYPE = 6,
   INVALID_HANDLE = 7,

   WRONG_INPUT = 10,

   DISTRIBUTED_TRAN_FAILED = 15,

   TPCC_DB_INFO_PARTIAL = 20,
   TPCC_DB_INFO_FAILED,

   TPCC_ERROR_BEGIN_NEWO = 110,

   TPCC_ERROR_DECL_NEWO_SEL_ITEM,
   TPCC_ERROR_OPEN_NEWO_SEL_ITEM,
   TPCC_ERROR_OPEN_DIST_NEWO_SEL_ITEM,
   TPCC_ERROR_FETCH_NEWO_SEL_ITEM,
   TPCC_ERROR_FETCH_DIST_NEWO_SEL_ITEM,
   TPCC_ERROR_PREP_NEWO_SEL_STCK,
   TPCC_ERROR_DECL_NEWO_SEL_STCK,
   TPCC_ERROR_OPEN_NEWO_SEL_STCK,
   TPCC_ERROR_OPEN_DIST_NEWO_SEL_STCK,
   TPCC_ERROR_FETCH_NEWO_SEL_STCK,
   TPCC_ERROR_FETCH_DIST_NEWO_SEL_STCK,
   TPCC_ERROR_NEWO_SELECT,
   TPCC_ERROR_NEWO_UPD_STCK,
   TPCC_ERROR_DIST_NEWO_UPD_STCK,
   TPCC_ERROR_NEWO_SELECT_2,
   TPCC_ERROR_DECL_NEWO_SEL_CUST,
   TPCC_ERROR_OPEN_NEWO_SEL_CUST,
   TPCC_ERROR_OPEN_DIST_NEWO_SEL_CUST,
   TPCC_ERROR_FETCH_NEWO_SEL_CUST,
   TPCC_ERROR_FETCH_DIST_NEWO_SEL_CUST,
   TPCC_ERROR_DECL_NEWO_SEL_DIST,
   TPCC_ERROR_OPEN_NEWO_SEL_DIST,
   TPCC_ERROR_OPEN_DIST_NEWO_SEL_DIST,
   TPCC_ERROR_FETCH_NEWO_SEL_DIST,
   TPCC_ERROR_FETCH_DIST_NEWO_SEL_DIST,
   TPCC_ERROR_PREP_NEWO_INS_OL,
   TPCC_ERROR_DECL_NEWO_INS_OL,
   TPCC_ERROR_OPEN_NEWO_INS_OL,
   TPCC_ERROR_OPEN_DIST_NEWO_INS_OL,
   TPCC_ERROR_PUT_NEWO_INS_OL,
   TPCC_ERROR_PUT_DIST_NEWO_INS_OL,
   TPCC_ERROR_DECL_NEWO_SEL_WARE,
   TPCC_ERROR_OPEN_NEWO_SEL_WARE,
   TPCC_ERROR_OPEN_DIST_NEWO_SEL_WARE,
   TPCC_ERROR_FETCH_NEWO_SEL_WARE,
   TPCC_ERROR_FETCH_DIST_NEWO_SEL_WARE,
   TPCC_ERROR_EXECUTE_NEWO_UPD_INS,
   TPCC_ERROR_UPDATE_NEWO_NEXT_OID,
   TPCC_ERROR_PREP_NEWO_INS,
   TPCC_ERROR_EXECUTE_DIST_NEWO_INS,
   TPCC_ERROR_EXECUTE_NEWO_COMMIT,
   TPCC_ERROR_ROLLBACK_NEWO,
   TPCC_ERROR_REMOTE_OL_SELECT,
   TPCC_ERROR_REMOTE_OL_UPDATE,

   TPCC_ERROR_OPEN_ORDS_CNT_CID = 200,
   TPCC_ERROR_FETCH_ORDS_CNT_CID,
   TPCC_ERROR_OPEN_ORDS_SEL_CLAST,
   TPCC_ERROR_FETCH_ORDS_SEL_CLAST,
   TPCC_ERROR_OPEN_ORDS_SEL_CID,
   TPCC_ERROR_FETCH_ORDS_SEL_CID,
   TPCC_ERROR_OPEN_ORDS_SEL_OLDORD,
   TPCC_ERROR_FETCH_ORDS_OLDORD,
   TPCC_ERROR_OPEN_ORDS_SEL_OL,
   TPCC_ERROR_FETCH_ORDS_SEL_OL,
   TPCC_ERROR_EXECUTE_ORDS_COMMIT,

   TPCC_ERROR_OPEN_DELIVERY_OLDEST_OID = 300,
   TPCC_ERROR_FETCH_DELIVERY_OLDEST_OID,
```

```
    TPCC_ERROR_EXECUTE_DELIVERY_COMMIT,
    TPCC_ERROR_OPEN_DELIVERY_SEL_ORD,
    TPCC_ERROR_FETCH_DELIVERY_SEL_ORD,
    TPCC_ERROR_OPEN_DELIVERY_SEL_SUM_OL,
    TPCC_ERROR_FETCH_DELIVERY_SEL_SUM_OL,
    TPCC_ERROR_EXECUTE_DELIVERY_EXEC_DVRY,
    TPCC_ERROR_SELECT_DELIVERY_ORDER_ID,
    TPCC_ERROR_SELECT_DELIVERY_CARRIER_ID,
    TPCC_ERROR_SELECT_DELIVERY_BALANCE,

    TPCC_ERROR_OPEN_STOCKLEVEL_SEL_OID = 400,
    TPCC_ERROR_FETCH_STOCKLEVEL_SEL_OID,
    TPCC_ERROR_OPEN_STOCKLEVEL_CNT_SID,
    TPCC_ERROR_FETCH_STOCKLEVEL_CNT_SID,
    TPCC_ERROR_OPEN_STOCKLEVEL_FIND,
    TPCC_ERROR_FETCH_STOCKLEVEL_FIND,
    TPCC_ERROR_EXECUTE_STOCKLEVEL_COMMIT,

    TPCC_ERROR_OPEN_PAYMENT_CNT_CID = 500,
    TPCC_ERROR_FETCH_PAYMENT_CNT_CID,
    TPCC_ERROR_OPEN_PAYMENT_SEL_CLAST,
    TPCC_ERROR_FETCH_PAYMENT_SEL_CLAST,
    TPCC_ERROR_OPEN_PAYMENT_SEL_CID,
    TPCC_ERROR_FETCH_PAYMENT_SEL_CID,
    TPCC_ERROR_DECL_PAYMENT_SEL_DIST,
    TPCC_ERROR_OPEN_PAYMENT_SEL_DIST,
    TPCC_ERROR_OPEN_DIST_PAYMENT_SEL_DIST,
    TPCC_ERROR_FETCH_PAYMENT_SEL_DIST,
    TPCC_ERROR_FETCH_DIST_PAYMENT_SEL_DIST,
    TPCC_ERROR_DECL_PAYMENT_SEL_WARE,
    TPCC_ERROR_OPEN_PAYMENT_SEL_WARE,
    TPCC_ERROR_OPEN_DIST_PAYMENT_SEL_WARE,
    TPCC_ERROR_FETCH_PAYMENT_SEL_WARE,
    TPCC_ERROR_FETCH_DIST_PAYMENT_SEL_WARE,
    TPCC_ERROR_EXECUTE_PAYMENT_UPD_CUST_LAST,
    TPCC_ERROR_EXECUTE_PAYMENT_UPD_CUST_ID,
    TPCC_ERROR_COMMIT_PAYMENT_UPD_CUST,
    TPCC_ERROR_SELECT_PAYMENT_W_YTD,
    TPCC_ERROR_SELECT_PAYMENT_D_YTD,
    TPCC_ERROR_BEGIN_PAYMENT,
    TPCC_ERROR_EXECUTE_PAYMENT_COMMIT,
    TPCC_ERROR_PAYMENT_UPD_CUST_BY_NAME,
    TPCC_ERROR_PAYMENT_UPD_CUST_BY_ID,
    TPCC_ERROR_PAYMENT_UPDATE_DIST,
    TPCC_ERROR_PAYMENT_UPDATE_WH,
    TPCC_ERROR_PAYMENT_INSERT_HISTORY,
    TPCC_ERROR_EXECUTE_PAYMENT_WH_DIST

} tpcc_rc_t;

typedef enum {
    TPCC_DEADLOCK_MSG = 10,
    TPCC_RETRY_MSG
} tpcc_msg_t;

#endif /* __TPCC_DATABUF_H__ */
```

databuf.h.new

```
/*
 *    databuf.h
 */

#ifndef __TPCC_DATABUF_H__
#define __TPCC_DATABUF_H__

#define I_NAME_LEN      24
#define I_DATA          50
#define W_NAME_LEN      10
#define ADDR_LEN        20
#define STATE_LEN       2
#define ZIP_LEN         9
#define DIST_INFO_LEN   24
#define S_DATA_LEN      50
#define D_NAME_LEN      10
#define H_DATA_LEN      24
#define CARRIER_LEN     2
#define C_LAST_LEN              17
#define C_MID_LEN       2
#define PHONE_LEN       16
#define CREDIT_LEN      2
/* #define C_DATA_LEN     500 */
#define BC_DTA_LEN              23

#define YEAR_TO_DATE   1
#define YEAR_TO_SECOND 2

#define MAX_STR_LEN     255
#define MAX_OL                  15

#ifndef TRUE
#define TRUE    1
#endif
#ifndef FALSE
#define FALSE   0
#endif

#define CANCEL          -1

/* #define DATETIME_LEN    19 */

#define D_PER_W         10

#define COLLECTOR1                      /* ctipper 5/3/95 */

#define ERR_BAD_ITEM_ID  1  /* copied from sql/tpcc.h */
#define RPC_ERROR               -2
#define SUCCESS_CODE                    0

#define CHAR_NULL       '\0'            /* strue 1/23/95 */

typedef enum {
    liberty_term,
    curses_term,
    emulator_term
} term_type_t;

typedef enum {
    TPCC_SUCCESS = 0,
    GOOD_INPUT = 0,

    INVALID_NEWO = 100,
    SQL_ERROR = 2,
    TRPC_ERROR = 3,
    DCE_ERROR = 4,
    NO_SUCH_LAST_NAME = 5,
    INVALID_TRAN_TYPE = 6,
    INVALID_HANDLE = 7,

    WRONG_INPUT = 10,

    DISTRIBUTED_TRAN_FAILED = 15,

    TPCC_DB_INFO_PARTIAL = 20,
    TPCC_DB_INFO_FAILED,

    TPCC_ERROR_BEGIN_NEWO = 110,

    TPCC_ERROR_DECL_NEWO_SEL_ITEM,
    TPCC_ERROR_OPEN_NEWO_SEL_ITEM,
```

```
    TPCC_ERROR_OPEN_DIST_NEWO_SEL_ITEM,
    TPCC_ERROR_FETCH_NEWO_SEL_ITEM,
    TPCC_ERROR_FETCH_DIST_NEWO_SEL_ITEM,
    TPCC_ERROR_PREP_NEWO_SEL_STCK,
    TPCC_ERROR_DECL_NEWO_SEL_STCK,
    TPCC_ERROR_OPEN_NEWO_SEL_STCK,
    TPCC_ERROR_OPEN_DIST_NEWO_SEL_STCK,
    TPCC_ERROR_FETCH_NEWO_SEL_STCK,
    TPCC_ERROR_FETCH_DIST_NEWO_SEL_STCK,
    TPCC_ERROR_NEWO_SELECT,
    TPCC_ERROR_NEWO_UPD_STCK,
    TPCC_ERROR_DIST_NEWO_UPD_STCK,
    TPCC_ERROR_NEWO_SELECT_2,
    TPCC_ERROR_DECL_NEWO_SEL_CUST,
    TPCC_ERROR_OPEN_NEWO_SEL_CUST,
    TPCC_ERROR_OPEN_DIST_NEWO_SEL_CUST,
    TPCC_ERROR_FETCH_NEWO_SEL_CUST,
    TPCC_ERROR_FETCH_DIST_NEWO_SEL_CUST,
    TPCC_ERROR_DECL_NEWO_SEL_DIST,
    TPCC_ERROR_OPEN_NEWO_SEL_DIST,
    TPCC_ERROR_OPEN_DIST_NEWO_SEL_DIST,
    TPCC_ERROR_FETCH_NEWO_SEL_DIST,
    TPCC_ERROR_FETCH_DIST_NEWO_SEL_DIST,
    TPCC_ERROR_PREP_NEWO_INS_OL,
    TPCC_ERROR_DECL_NEWO_INS_OL,
    TPCC_ERROR_OPEN_NEWO_INS_OL,
    TPCC_ERROR_OPEN_DIST_NEWO_INS_OL,
    TPCC_ERROR_PUT_NEWO_INS_OL,
    TPCC_ERROR_PUT_DIST_NEWO_INS_OL,
    TPCC_ERROR_DECL_NEWO_SEL_WARE,
    TPCC_ERROR_OPEN_NEWO_SEL_WARE,
    TPCC_ERROR_OPEN_DIST_NEWO_SEL_WARE,
    TPCC_ERROR_FETCH_NEWO_SEL_WARE,
    TPCC_ERROR_FETCH_DIST_NEWO_SEL_WARE,
    TPCC_ERROR_EXECUTE_NEWO_UPD_INS,
    TPCC_ERROR_UPDATE_NEWO_NEXT_OID,
    TPCC_ERROR_PREP_NEWO_INS,
    TPCC_ERROR_EXECUTE_DIST_NEWO_INS,
    TPCC_ERROR_EXECUTE_NEWO_COMMIT,
    TPCC_ERROR_ROLLBACK_NEWO,
    TPCC_ERROR_REMOTE_OL_SELECT,
    TPCC_ERROR_REMOTE_OL_UPDATE,

    TPCC_ERROR_OPEN_ORDS_CNT_CID = 200,
    TPCC_ERROR_FETCH_ORDS_CNT_CID,
    TPCC_ERROR_OPEN_ORDS_SEL_CLAST,
    TPCC_ERROR_FETCH_ORDS_SEL_CLAST,
    TPCC_ERROR_OPEN_ORDS_SEL_CID,
    TPCC_ERROR_FETCH_ORDS_SEL_CID,
    TPCC_ERROR_OPEN_ORDS_SEL_OLDORD,
    TPCC_ERROR_FETCH_ORDS_OLDORD,
    TPCC_ERROR_OPEN_ORDS_SEL_OL,
    TPCC_ERROR_FETCH_ORDS_SEL_OL,
    TPCC_ERROR_EXECUTE_ORDS_COMMIT,

    TPCC_ERROR_OPEN_DELIVERY_OLDEST_OID = 300,
    TPCC_ERROR_FETCH_DELIVERY_OLDEST_OID,
    TPCC_ERROR_EXECUTE_DELIVERY_COMMIT,
    TPCC_ERROR_OPEN_DELIVERY_SEL_ORD,
    TPCC_ERROR_FETCH_DELIVERY_SEL_ORD,
    TPCC_ERROR_OPEN_DELIVERY_SEL_SUM_OL,
    TPCC_ERROR_FETCH_DELIVERY_SEL_SUM_OL,
    TPCC_ERROR_EXECUTE_DELIVERY_EXEC_DVRY,
    TPCC_ERROR_SELECT_DELIVERY_ORDER_ID,
    TPCC_ERROR_SELECT_DELIVERY_CARRIER_ID,
    TPCC_ERROR_SELECT_DELIVERY_BALANCE,

    TPCC_ERROR_OPEN_STOCKLEVEL_SEL_OID = 400,
    TPCC_ERROR_FETCH_STOCKLEVEL_SEL_OID,
    TPCC_ERROR_OPEN_STOCKLEVEL_CNT_SID,
    TPCC_ERROR_FETCH_STOCKLEVEL_CNT_SID,
    TPCC_ERROR_OPEN_STOCKLEVEL_FIND,
    TPCC_ERROR_FETCH_STOCKLEVEL_FIND,
    TPCC_ERROR_EXECUTE_STOCKLEVEL_COMMIT,

    TPCC_ERROR_OPEN_PAYMENT_CNT_CID = 500,
    TPCC_ERROR_FETCH_PAYMENT_CNT_CID,
    TPCC_ERROR_OPEN_PAYMENT_SEL_CLAST,
    TPCC_ERROR_FETCH_PAYMENT_SEL_CLAST,
    TPCC_ERROR_OPEN_PAYMENT_SEL_CID,
    TPCC_ERROR_FETCH_PAYMENT_SEL_CID,
    TPCC_ERROR_DECL_PAYMENT_SEL_DIST,
    TPCC_ERROR_OPEN_PAYMENT_SEL_DIST,
    TPCC_ERROR_OPEN_DIST_PAYMENT_SEL_DIST,
    TPCC_ERROR_FETCH_PAYMENT_SEL_DIST,
    TPCC_ERROR_FETCH_DIST_PAYMENT_SEL_DIST,
    TPCC_ERROR_DECL_PAYMENT_SEL_WARE,
    TPCC_ERROR_OPEN_PAYMENT_SEL_WARE,
    TPCC_ERROR_OPEN_DIST_PAYMENT_SEL_WARE,
    TPCC_ERROR_FETCH_PAYMENT_SEL_WARE,
    TPCC_ERROR_FETCH_DIST_PAYMENT_SEL_WARE,
    TPCC_ERROR_EXECUTE_PAYMENT_UPD_CUST_LAST,
    TPCC_ERROR_EXECUTE_PAYMENT_UPD_CUST_ID,
    TPCC_ERROR_COMMIT_PAYMENT_UPD_CUST,
    TPCC_ERROR_SELECT_PAYMENT_W_YTD,
    TPCC_ERROR_SELECT_PAYMENT_D_YTD,
    TPCC_ERROR_BEGIN_PAYMENT,
    TPCC_ERROR_EXECUTE_PAYMENT_COMMIT,
    TPCC_ERROR_PAYMENT_UPD_CUST_BY_NAME,
    TPCC_ERROR_PAYMENT_UPD_CUST_BY_ID,
    TPCC_ERROR_PAYMENT_UPDATE_DIST,
    TPCC_ERROR_PAYMENT_UPDATE_WH,
    TPCC_ERROR_PAYMENT_INSERT_HISTORY,
    TPCC_ERROR_EXECUTE_PAYMENT_WH_DIST

} tpcc_rc_t;

typedef enum {
    TPCC_DEADLOCK_MSG = 10,
    TPCC_RETRY_MSG
} tpcc_msg_t;

#endif /* __TPCC_DATABUF_H__ */
```

db_dblib_dll.dsp

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

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

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

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

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

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

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

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

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

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

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

!ENDIF

# Begin Target

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

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

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

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

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

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

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

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

delivery.h

```
#ifndef TRANSARC_delivery_h
#define TRANSARC_delivery_h

#include <trpc/trpc.h>
#include "_delivery.h"

#include <encina/c_prologue.h>

#if defined(BUILDDLL)
#define DLLEXPORT __declspec( dllexport )
#else
#define DLLEXPORT extern
#endif

#ifndef ENCINA_STUB_CALLING
#define ENCINA_STUB_CALLING ENCINA_RPC_CALLING
#endif

#define delivery_v1_0_c_ifspec  _delivery_v1_0_c_ifspec
#define delivery_v1_0_s_ifspec  _delivery_v1_0_s_ifspec

typedef struct delivery_v1_0_epv {
void (ENCINA_STUB_CALLING *impTPCCDelivery) (
#ifdef IDL_PROTOTYPES

                        idl_long_int length,
                        idl_char *dataP,
                        data_header *headerP,
                        trpc_status_t *trpcStatus
#endif
);

} delivery_v1_0_epv_t;

DLLEXPORT void ENCINA_STUB_CALLING impTPCCDelivery (
#ifdef IDL_PROTOTYPES

                        idl_long_int length,
                        idl_char *dataP,
                        data_header *headerP,
                        trpc_status_t *trpcStatus
#endif
);

trpc_handle_t           ENCINA_CALLING mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
            mon_handle_t        handle,
            trpc_tranInfo_t     *tranInfoP,
            trpc_ifSpec_t *ifSpecP
#endif
            );

void        ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
            mon_handle_t        handle,
            trpc_handle_t       trpcHandle,
            trpc_tranInfo_t     *tranInfoP,
            trpc_ifSpec_t *ifSpecP
#endif
            );

trpc_handle_t           ENCINA_CALLING mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
            mon_handle_t        handle,
            trpc_tranInfo_t     *tranInfoP,
            trpc_ifSpec_t *ifSpecP
#endif
            );

void        ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
            mon_handle_t        handle,
            trpc_handle_t       trpcHandle,
            trpc_tranInfo_t     *tranInfoP,
            trpc_ifSpec_t *ifSpecP
#endif
            );

extern delivery_v1_0_epv_t      delivery_v1_0_client_epv;
extern _delivery_v1_0_epv_t     delivery_v1_0_manager_epv;
extern rpc_mgr_epv_t            delivery_v1_0_mgr_epv;

#include <encina/c_epilogue.h>
#endif /* TRANSARC_delivery_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
 */

#pragma once

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

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

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

#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

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);
        }

        ~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,
	};

	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 "";}		// TODO: need to code
error text
};

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

	int ErrorType() { return ERR_TYPE_MEMORY;}
	int ErrorNum() { return 0;}
	char *ErrorText() { return "";}		// TODO: need to code
error text
};
```

install.c

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

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

#include "resource.h"

#define	WM_INITTEXT		WM_USER+100

HICON		hIcon;
HINSTANCE	hInst;

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

// TPC-C registry settings
TPCCREGISTRYDATA	Reg;

static	int		iPoolThreadLimit;
static	int		iThreadTimeout;
static	int		iListenBackLog;
static	int		iAcceptExOutstanding;

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


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

BOOL install_com(char *szDllPath);

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

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

	hInst = hInstance;

	InitCommonControls();

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

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

	DestroyIcon(hIcon);
	return 0;
}

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

	switch(uMsg)
	{
		case WM_INITDIALOG:
			hFont = CreateFont(-12, 0, 0, 0, 400, 0, 0, 0, 0, 0, 0, 0, 0, "Arial");
			SendMessage( GetDlgItem(hwnd, IDR_LICENSE1), WM_SETFONT, (WPARAM)hFont, MAKELPARAM(0, 0) );
			PostMessage(hwnd, WM_INITTEXT, (WPARAM)0, (LPARAM)0);
			return TRUE;
		case WM_INITTEXT:
			hResInfo = FindResource(hInst, MAKEINTRESOURCE(IDR_LICENSE1), "LICENSE");
			dwSize = SizeofResource(hInst, hResInfo);
			hRes = LoadResource(hInst, hResInfo );
			pSrc = (BYTE *)LockResource(hRes);
			pDst = (unsigned char *)malloc(dwSize+1);
			if ( pDst )
			{
				memcpy(pDst, pSrc, dwSize);
				pDst[dwSize] = 0;
				SetDlgItemText(hwnd, IDC_LICENSE, (const char *)pDst);
				free(pDst);
			}
			else
				SetDlgItemText(hwnd, IDC_LICENSE, (const char *)pSrc);
			return TRUE;
		case WM_DESTROY:
			DeleteObject(hFont);
			return TRUE;
		case WM_COMMAND:
			if ( wParam == IDOK )
				EndDialog(hwnd, TRUE);
			if ( wParam == IDCANCEL )
				EndDialog(hwnd, FALSE);
		default:
			break;
	}
```

```
	return FALSE;
}

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

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

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

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

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

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

			ReadTPCCRegistrySettings( &Reg );
			ReadRegistrySettings();

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

			wsprintf(szTmp, "Version %d.%2.2d.%3.3d",
versionExeMS, versionExeMM, versionExeLS);
			SetDlgItemText(hwnd, IDC_VERSION,
szTmp);

			SetDlgItemText(hwnd, IDC_PATH,
szDllPath);

			SetDlgItemText(hwnd, ED_DB_SERVER,
Reg.szDbServer);
			SetDlgItemText(hwnd, ED_DB_USER_ID,
Reg.szDbUser);
			SetDlgItemText(hwnd, ED_DB_PASSWORD,
Reg.szDbPassword);
			SetDlgItemText(hwnd, ED_DB_NAME,
Reg.szDbName);

			SetDlgItemInt(hwnd, ED_THREADS,
Reg.dwNumberOfDeliveryThreads, FALSE);
			SetDlgItemInt(hwnd,
ED_MAXCONNECTION, Reg.dwMaxConnections, FALSE);
			SetDlgItemInt(hwnd, ED_MAXDELIVERIES,
Reg.dwMaxPendingDeliveries, FALSE);
			SetDlgItemInt(hwnd,
ED_IIS_MAX_THEAD_POOL_LIMIT, iPoolThreadLimit, FALSE);
			SetDlgItemInt(hwnd,
ED_IIS_THREAD_TIMEOUT, iThreadTimeout, FALSE);
			SetDlgItemInt(hwnd,
ED_IIS_LISTEN_BACKLOG, iListenBackLog, FALSE);
			SetDlgItemInt(hwnd,
ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE, iAcceptExOutstanding,
FALSE);

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

			// check OS version level for COM.  Must be at
least Windows 2000
			VI.dwOSVersionInfoSize = sizeof(VI);
			GetVersionEx( &VI );
			if (VI.dwMajorVersion < 5)
			{
				HWND hDlg = GetDlgItem( hwnd,
IDC_TM_MTS );
				EnableWindow( hDlg, 0 );	//
disable COM option
				if (Reg.eTxnMon == COM)
					Reg.eTxnMon = None;
			}

			CheckDlgButton(hwnd, IDC_TM_NONE,   0);
			CheckDlgButton(hwnd, IDC_TM_TUXEDO,
0);
			CheckDlgButton(hwnd, IDC_TM_MTS,    0);
			CheckDlgButton(hwnd, IDC_TM_ENCINA,
0);
			switch (Reg.eTxnMon)
```

```
                                {
                                case None:
                                        CheckDlgButton(hwnd,
IDC_TM_NONE,   1);
                                        break;
                                case TUXEDO:
                                        CheckDlgButton(hwnd,
IDC_TM_TUXEDO, 1);
                                        break;
                                case ENCINA:
                                        CheckDlgButton(hwnd,
IDC_TM_ENCINA, 1);
                                        break;
                                case COM:
                                        CheckDlgButton(hwnd,
IDC_TM_MTS,        1);
                                        break;
                                }

                                return TRUE;
                        case WM_PAINT:
                                if ( IsIconic(hwnd) )
                                {
                                        BeginPaint(hwnd, &ps);
                                        DrawIcon(ps.hdc, 0, 0, hIcon);
                                        EndPaint(hwnd, &ps);
                                        return TRUE;
                                }
                                break;
                        case WM_COMMAND:
                                if ( HIWORD(wParam) == BN_CLICKED )
                                {
                                        switch( LOWORD(wParam) )
                                        {
                                                case IDC_DBLIB:
                                                        return
TRUE;
                                                case IDC_ODBC:
                                                        return
TRUE;
                                                case IDOK:

ProcessOK(hwnd, szDllPath);
                                                        return
TRUE;
                                                case IDCANCEL:

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

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

        char    szFullName[256];
        char    szErrTxt[128];

        // read settings from dialog
        Reg.dwNumberOfDeliveryThreads  = GetDlgItemInt(hwnd,
ED_THREADS, &d, FALSE);
        Reg.dwMaxConnections = GetDlgItemInt(hwnd,
ED_MAXCONNECTION, &d, FALSE);
        Reg.dwMaxPendingDeliveries = GetDlgItemInt(hwnd,
ED_MAXDELIVERIES, &d, FALSE);

        GetDlgItemText(hwnd, ED_DB_SERVER, Reg.szDbServer,
sizeof(Reg.szDbServer));
        GetDlgItemText(hwnd, ED_DB_USER_ID, Reg.szDbUser,
sizeof(Reg.szDbUser));
        GetDlgItemText(hwnd, ED_DB_PASSWORD, Reg.szDbPassword,
sizeof(Reg.szDbPassword));
        GetDlgItemText(hwnd, ED_DB_NAME, Reg.szDbName,
sizeof(Reg.szDbName));

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

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

        iPoolThreadLimit = GetDlgItemInt(hwnd,
ED_IIS_MAX_THEAD_POOL_LIMIT, &d, FALSE);
        iThreadTimeout = GetDlgItemInt(hwnd,
ED_IIS_THREAD_TIMEOUT,   &d, FALSE);
        iListenBackLog = GetDlgItemInt(hwnd,
ED_IIS_LISTEN_BACKLOG, &d, FALSE);
        iAcceptExOutstanding = GetDlgItemInt(hwnd,
ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE, &d, FALSE);


        ShowWindow(hwnd, SW_HIDE);
        hDlg = CreateDialog(hInst,
MAKEINTRESOURCE(IDD_DIALOG3), hwnd, CopyDlgProc);
        ShowWindow(hDlg, SW_SHOWNA);
        UpdateDialog(hDlg);


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

        // update registry
        SetDlgItemText(hDlg, IDC_STATUS, "Updating Registry.");
```

```
		SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0,
0);
		UpdateDialog(hDlg);
		WriteRegistrySettings(szDllPath);

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

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

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

		Sleep(100);

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

		EndDialog(hwnd, rc);
		return;
}

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

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

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

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

			RegCloseKey(hKey);
		}

		if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters", 0, KEY_READ,
&hKey) == ERROR_SUCCESS )
		{
			size = sizeof(iAcceptExOutstanding);
			if ( RegQueryValueEx(hKey, "AcceptExOutstanding", 0,
&type, (char *)&iAcceptExOutstanding, &size) == ERROR_SUCCESS )
				if ( !iAcceptExOutstanding )
					iAcceptExOutstanding = 40;

			RegCloseKey(hKey);
		}
}

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

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

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

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

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

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

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

                RegFlushKey(hKey);
                RegCloseKey(hKey);
        }

        if ( (iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Parameters", 0, NULL,
REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, NULL, &hKey,
&dwDisposition)) == ERROR_SUCCESS )
        {
                RegSetValueEx(hKey, "PoolThreadLimit", 0,
REG_DWORD, (char *)&iPoolThreadLimit, sizeof(iPoolThreadLimit));
                RegSetValueEx(hKey, "ThreadTimeout", 0,
REG_DWORD, (char *)&iThreadTimeout, sizeof(iThreadTimeout));
                RegSetValueEx(hKey, "ListenBackLog", 0,
REG_DWORD, (char *)&iListenBackLog, sizeof(iListenBackLog));

                RegFlushKey(hKey);
                RegCloseKey(hKey);
        }

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

                RegFlushKey(hKey);
                RegCloseKey(hKey);
        }

        return;
}

BOOL CALLBACK CopyDlgProc(HWND hwnd, UINT uMsg, WPARAM
wParam, LPARAM lParam)
{
        if ( uMsg == WM_INITDIALOG )
        {
                SendDlgItemMessage(hwnd, IDC_PROGRESS1,
PBM_SETRANGE, 0, MAKELPARAM(0, 15));
                SendDlgItemMessage(hwnd, IDC_PROGRESS1,
PBM_SETSTEP, (WPARAM)1, 0);
                return TRUE;
        }
        return FALSE;
}

BOOL RegisterDLL(char *szFileName)
{
        HINSTANCE       hLib;
        FARPROC         lpDllEntryPoint;

        hLib = LoadLibrary(szFileName);
        if ( hLib == NULL )
                return FALSE;
        // Find the entry point.
        lpDllEntryPoint = GetProcAddress(hLib, "DllRegisterServer");
        if (lpDllEntryPoint != NULL)
        {
                return ((*lpDllEntryPoint)() == S_OK);
        }
        else
                return FALSE;           //unable to locate entry point
}

BOOL FileFromResource( char *szResourceName, int iResourceId, char
*szDllPath, char *szFileName )
{
        HGLOBAL                         hDLL;
        HRSRC                   hResInfo;
        HANDLE                  hFile;
        DWORD                   dwSize;
        BYTE                    *pSrc;
        DWORD                   d;
        char                    szFullName[256];

        hResInfo = FindResource(hInst,
MAKEINTRESOURCE(iResourceId), szResourceName);

        strcpy(szFullName, szDllPath);
        strcat(szFullName, szFileName);

        dwSize = SizeofResource(hInst, hResInfo);
        hDLL = LoadResource(hInst, hResInfo );
        pSrc = (BYTE *)LockResource(hDLL);
        remove(szFullName);

        if ( !(hFile = CreateFile(szFullName, GENERIC_WRITE, 0, NULL,
CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL)) )
                return FALSE;

        if ( !WriteFile(hFile, pSrc, dwSize, &d, NULL) )
                return FALSE;

        CloseHandle(hFile);

        UnlockResource(hDLL);
        FreeResource(hDLL);
        return TRUE;
}

static int CopyFiles(HWND hDlg, char *szDllPath)
{
        BOOL                    bSvcRunning;

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

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

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

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

        // install tpcc_dblib.dll
        strcpy( szLastFileName, "tpcc_dblib.dll" );
```

```
	if (!FileFromResource( "DBLIB_DLL", IDR_DBLIB_DLL,
szDllPath, szLastFileName ))
		return 0;
	SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0,
0);
	UpdateDialog(hDlg);

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

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

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

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

	// install tpcc_com_ps.dll
	strcpy( szLastFileName, "tpcc_com_ps.dll" );
	if (!FileFromResource( "COM_PS_DLL", IDR_COMPS_DLL,
szDllPath, szLastFileName ))
		return 0;
	SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0,
0);
	UpdateDialog(hDlg);

	// install tpcc_com_all.dll
	strcpy( szLastFileName, "tpcc_com_all.dll" );
	if (!FileFromResource( "COM_ALL_DLL", IDR_COMALL_DLL,
szDllPath, szLastFileName ))
		return 0;
	SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0,
0);
	UpdateDialog(hDlg);

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

	return 1;
}

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

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

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

		RegCloseKey(hKey);
	}

	return bRc;
}

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

	versionDllMS = 0;
	versionDllLS = 0;
	if ( _access(szDLLPath, 00) == 0 )
	{
		dwSize = GetFileVersionInfoSize(szDLLPath, &d);
		if ( dwSize )
		{
			ptr = (char *)malloc(dwSize);
			GetFileVersionInfo(szDLLPath, 0, dwSize,
ptr);
			VerQueryValue(ptr, "\\",&vs, &dwBytes);
```

```
                        versionDllMS = vs->dwProductVersionMS;
                        versionDllLS = vs->dwProductVersionLS;
                        free(ptr);
                }
        }

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

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

static BOOL CheckWWWebService(void)
{
        SC_HANDLE               schSCManager;
        SC_HANDLE               schService;
        SERVICE_STATUS  ssStatus;

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

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

        if ( !ControlService(schService, SERVICE_CONTROL_STOP,
&ssStatus) )
                goto ServiceNotRunning;
        //start Service pending, Check the status until the service is running.
        if (! QueryServiceStatus(schService, &ssStatus) )
                goto ServiceNotRunning;

        CloseServiceHandle(schService);
        return TRUE;

ServiceNotRunning:

        CloseServiceHandle(schService);
        return FALSE;
}

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

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

        if (! StartService(schService, 0, NULL) )
                goto StartWWWebErr;
        //start Service pending, Check the status until the service is running.
        if (! QueryServiceStatus(schService, &ssStatus) )
                goto StartWWWebErr;
        while( ssStatus.dwCurrentState != SERVICE_RUNNING)
        {

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

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

        CloseServiceHandle(schService);
        return TRUE;

StartWWWebErr:
        CloseServiceHandle(schService);
        return FALSE;
}

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

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

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

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

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

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

        CloseServiceHandle(schService);
```

```
        return TRUE;

StopWWWebErr:
        CloseServiceHandle(schService);
        return FALSE;
}

static void UpdateDialog(HWND hDlg)
{
        MSG msg;

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

install.dsp

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

# TARGTYPE "Win32 (x86) Application" 0x0101

CFG=install - 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 "install.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 "install.mak" CFG="install - Win32 Release"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "install - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "install - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

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

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir ".\Release"
# PROP BASE Intermediate_Dir ".\Release"
# PROP BASE Target_Dir "."
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir "."
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
"_WINDOWS" /YX /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
"_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib
odbccp32.lib /nologo /subsystem:windows /machine:I386
# ADD LINK32 version.lib comctl32.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib
odbc32.lib odbccp32.lib /nologo /subsystem:windows /machine:I386
/out:"..\bin\install.exe"

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir ".\Debug"
# PROP BASE Intermediate_Dir ".\Debug"
# PROP BASE Target_Dir "."
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir "."
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG"
/D "_WINDOWS" /YX /c
# ADD CPP /nologo /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D
"_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib
odbccp32.lib /nologo /subsystem:windows /debug /machine:I386
# ADD LINK32 version.lib comctl32.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib
odbc32.lib odbccp32.lib /nologo /subsystem:windows /debug /machine:I386
/out:"..\bin\install.exe"

!ENDIF

# Begin Target

# Name "install - Win32 Release"
# Name "install - Win32 Debug"
# Begin Group "Source Files"

# PROP Default_Filter "cpp;c;cxx;rc;def;r;odl;hpj;bat;for;f90"
# Begin Source File

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

SOURCE=.\src\install.rc
# ADD BASE RSC /l 0x409 /i "src"
# ADD RSC /l 0x409 /i "src" /i "..\src"
```

```
# End Source File
# Begin Source File

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

# PROP Default_Filter "h;hpp;hxx;hm;inl;fi;fd"
# End Group
# Begin Group "Resource Files"

# PROP Default_Filter "ico;cur;bmp;dlg;rc2;rct;bin;cnt;rtf;gif;jpg;jpeg;jpe"
# Begin Source File

SOURCE=.\SRC\ICON1.ICO
# End Source File
# Begin Source File

SOURCE=.\SRC\ICON2.ICO
# End Source File
# End Group
# Begin Source File

SOURCE=.\SRC\LICENSE.TXT
# End Source File
# Begin Source File

SOURCE=..\isapi_dll\bin\tpcc.dll
# End Source File
# Begin Source File

SOURCE=..\tm_com_dll\bin\tpcc_com.dll
# End Source File
# Begin Source File

SOURCE=..\tpcc_com_all\bin\tpcc_com_all.dll
# End Source File
# Begin Source File

SOURCE=..\tpcc_com_ps\bin\tpcc_com_ps.dll
# End Source File
# Begin Source File

SOURCE=..\db_dblib_dll\bin\tpcc_dblib.dll
# End Source File
# Begin Source File

SOURCE=..\db_odbc_dll\bin\tpcc_odbc.dll
# End Source File
# Begin Source File

SOURCE=..\tm_tuxedo_dll\bin\tpcc_tuxedo.dll
# End Source File
# Begin Source File

SOURCE=..\tuxapp\bin\tuxapp.exe
# End Source File
# End Target
# End Project
```

install.h

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

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

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

install.rc

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

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

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

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

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

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

IDD_DIALOG1 DIALOGEX 0, 0, 219, 351
STYLE DS_MODALFRAME | DS_CENTER | WS_MINIMIZEBOX |
WS_POPUP | WS_CAPTION |
    WS_SYSMENU
CAPTION "TPC-C Web Client Installation Utility"
```

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


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

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

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


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

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

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

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

```
    TOPMARGIN, 7
    BOTTOMMARGIN, 33
  END

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


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

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

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

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

#endif    // APSTUDIO_INVOKED


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

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

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

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

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

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

#endif    // !_MAC


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

IDR_LICENSE1            LICENSE DISCARDABLE     "license.txt"

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

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

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

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

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

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

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

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

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

```
//
// COM_DLL
//

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

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

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

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

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



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


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

install_com.cpp

```
/*          FILE:                   INSTALL_COM.CPP
 *                                          Microsoft TPC-C Kit Ver. 4.20.000
 *                                          Copyright Microsoft, 1999
 *          All Rights Reserved
 *
 *                                          not audited
 *
 *          PURPOSE:                installation code for COM application for
TPC-C Web Kit
 *          Contact:    Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                      4.20.000 - first version
 */

#define _WIN32_WINNT 0x0500

#include <comdef.h>
#include <comadmin.h>
#include <stdio.h>
#include <tchar.h>

extern "C"
{
          BOOL install_com(char *szDllPath);
}

BOOL install_com(char *szDllPath)
{
          ICOMAdminCatalog*pCOMAdminCat = NULL;
          ICatalogCollection*   pCatalogCollectionApp    = NULL;
          ICatalogCollection*   pCatalogCollectionCo     = NULL;
          ICatalogCollection*   pCatalogCollectionItf    = NULL;
          ICatalogCollection*   pCatalogCollectionMethod = NULL;

          ICatalogObject*               pCatalogObjectApp    = NULL;
          ICatalogObject*               pCatalogObjectCo     = NULL;
          ICatalogObject*               pCatalogObjectItf    = NULL;
          ICatalogObject*               pCatalogObjectMethod = NULL;

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


          CoInitializeEx(NULL, COINIT_MULTITHREADED);

          HRESULT hr = CoCreateInstance(CLSID_COMAdminCatalog,

          NULL,

          CLSCTX_INPROC_SERVER,

          IID_ICOMAdminCatalog,

          (void**) &pCOMAdminCat);

          if (!SUCCEEDED(hr)) goto Error;

          bstrTemp = "Applications";

          // Attempt to connect to "Applications" in the Catalog
          hr = pCOMAdminCat->GetCollection(bstrTemp,

          (IDispatch**) &pCatalogCollectionApp);
          if (!SUCCEEDED(hr)) goto Error;

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

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

          // iterate through applications to delete existing "TPC-C" application
(if any)
          while (lCount > 0)
          {
                    hr = pCatalogCollectionApp->get_Item(lCount - 1,
(IDispatch**) &pCatalogObjectApp);
                    if (!SUCCEEDED(hr)) goto Error;

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

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

```
                }
        }

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

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

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

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

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

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

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

        pCatalogObjectApp->Release();
        pCatalogObjectApp = NULL;

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

        hr = pCOMAdminCat->InstallComponent(bstrTemp,

                                bstrTemp2,

                                bstrTemp3,

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


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

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

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

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

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

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

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

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

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

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



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

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

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

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

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

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

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

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

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

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

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

                                        pCatalogObjectMethod->Release();
                                        pCatalogObjectMethod = NULL;

                                        lCountMethod--;
                                }

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

                                pCatalogObjectItf->Release();
                                pCatalogObjectItf = NULL;

                                lCountItf--;

                        }


                        pCatalogObjectCo->Release();
                        pCatalogObjectCo = NULL;

                        lCountCo--;
                }


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

                pCatalogCollectionApp->Release();
                pCatalogCollectionApp = NULL;

                pCatalogCollectionCo->Release();
                pCatalogCollectionCo = NULL;

                pCatalogCollectionItf->Release();
                pCatalogCollectionItf = NULL;

                pCatalogCollectionMethod->Release();
                pCatalogCollectionMethod = NULL;


Error:
                CoUninitialize();

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

                NULL,

                hr,

                MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),

                (LPTSTR) &lpBuf,

                0,

                NULL);
//                      _tprintf(__T("Error adding components.  HRESULT:
0x%x\n%s"), hr, lpBuf);
                        return TRUE;
                }
                else
                        return FALSE;
}
```

isapi_dll_resource.h

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

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

isapi_dll.dsp

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

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

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

```
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "isapi_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link
Library")
!MESSAGE "isapi_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link
Library")
!MESSAGE "isapi_dll - Win32 IceCAP" (based on "Win32 (x86) Dynamic-Link
Library")
!MESSAGE

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

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

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

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

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

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

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

!ENDIF

# Begin Target

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

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

SOURCE=.\src\tpcc.cpp
```

```
# End Source File
# Begin Source File

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

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

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

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

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

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

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

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

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

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

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

methods.h

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


enum COMPONENT_ERROR
{
        ERR_MISSING_REGISTRY_ENTRIES = 1,
        ERR_LOADDLL_FAILED,
        ERR_GETPROCADDR_FAILED,
        ERR_UNKNOWN_DB_PROTOCOL
};

class CCOMPONENT_ERR : public CBaseErr
{
        public:
                CCOMPONENT_ERR(COMPONENT_ERROR Err)
                {
                        m_Error = Err;
                        m_szTextDetail = NULL;
                        m_SystemErr = 0;
                        m_szErrorText = NULL;
                };

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

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

                COMPONENT_ERROR         m_Error;
                char                    *m_szTextDetail;
                char                    *m_szErrorText;
                DWORD                   m_SystemErr;

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


static void WriteMessageToEventLog(LPTSTR lpszMsg);


/////////////////////////////////////////////////////////////////////////////
// CTPCC_Common
class CTPCC_Common :
        public ITPCC,
        public IObjectControl,
        public IObjectConstruct,
        public CComObjectRootEx<CComSingleThreadModel>
{
public:
BEGIN_COM_MAP(CTPCC_Common)
        COM_INTERFACE_ENTRY(ITPCC)
        COM_INTERFACE_ENTRY(IObjectControl)
        COM_INTERFACE_ENTRY(IObjectConstruct)
END_COM_MAP()

        CTPCC_Common();
        ~CTPCC_Common();

// ITPCC
public:
        HRESULT __stdcall NewOrder(             int* iSize, UCHAR**
txn);
        HRESULT __stdcall Payment(              int* iSize, UCHAR**
txn);
```

```
        HRESULT __stdcall Delivery(         int* iSize, UCHAR**
txn) {return E_NOTIMPL;}
        HRESULT __stdcall StockLevel(int* iSize, UCHAR** txn);
        HRESULT __stdcall OrderStatus(int* iSize, UCHAR** txn);

        HRESULT __stdcall CallSetComplete();

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

// IObjectConstruct
        STDMETHODIMP Construct(IDispatch * pUnk);

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

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

};

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

BEGIN_COM_MAP(CTPCC)
        COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

};


/////////////////////////////////////////////////////////////////////////////
// CNewOrder
class CNewOrder :
        public CTPCC_Common,
        public CComCoClass<CNewOrder, &CLSID_NewOrder>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_NEWORDER)

BEGIN_COM_MAP(CNewOrder)
        COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
//      HRESULT __stdcall NewOrder(         int* iSize, UCHAR**
txn) {return E_NOTIMPL;}
        HRESULT __stdcall Payment(          int* iSize, UCHAR**
txn) {return E_NOTIMPL;}
        HRESULT __stdcall StockLevel(int* iSize, UCHAR** txn) {return
E_NOTIMPL;}
        HRESULT __stdcall OrderStatus(int* iSize, UCHAR** txn) {return
E_NOTIMPL;}
};


        /////////////////////////////////////////////////////////////////////////////
// COrderStatus
class COrderStatus :
        public CTPCC_Common,
        public CComCoClass<COrderStatus, &CLSID_OrderStatus>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_ORDERSTATUS)

BEGIN_COM_MAP(COrderStatus)
        COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
        HRESULT __stdcall NewOrder(         int* iSize, UCHAR**
txn) {return E_NOTIMPL;}
        HRESULT __stdcall Payment(          int* iSize, UCHAR**
txn) {return E_NOTIMPL;}
        HRESULT __stdcall StockLevel(int* iSize, UCHAR** txn) {return
E_NOTIMPL;}
//      HRESULT __stdcall OrderStatus(int* iSize, UCHAR** txn) {return
E_NOTIMPL;}
};


/////////////////////////////////////////////////////////////////////////////
// CPayment
class CPayment :
        public CTPCC_Common,
        public CComCoClass<CPayment, &CLSID_Payment>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_PAYMENT)

BEGIN_COM_MAP(CPayment)
        COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
        HRESULT __stdcall NewOrder(         int* iSize, UCHAR**
txn) {return E_NOTIMPL;}
//      HRESULT __stdcall Payment(          int* iSize, UCHAR**
txn) {return E_NOTIMPL;}
        HRESULT __stdcall StockLevel(int* iSize, UCHAR** txn) {return
E_NOTIMPL;}
        HRESULT __stdcall OrderStatus(int* iSize, UCHAR** txn) {return
E_NOTIMPL;}
};


/////////////////////////////////////////////////////////////////////////////
// CStockLevel
class CStockLevel :
        public CTPCC_Common,
        public CComCoClass<CStockLevel, &CLSID_StockLevel>
```

```
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_STOCKLEVEL)

BEGIN_COM_MAP(CStockLevel)
        COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
        HRESULT __stdcall NewOrder(             int* iSize, UCHAR**
txn) {return E_NOTIMPL;}
        HRESULT __stdcall Payment(              int* iSize, UCHAR**
txn) {return E_NOTIMPL;}
//      HRESULT __stdcall StockLevel(int* iSize, UCHAR** txn) {return
E_NOTIMPL;}
        HRESULT __stdcall OrderStatus(int* iSize, UCHAR** txn) {return
E_NOTIMPL;}
};
```

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  *perfClntDataInit();
static void read_mon_environment(void);

static char *cellName;
static int envRetrieval = 0;
static int useSecurity = FALSE;
static CRITICAL_SECTION    init_lock;
static total_tran_count_t  *pClientInfo=NULL;  /* keep stats for the client process
*/
static num_active_threads = 0;
static int iStatsFrequency = 1;
FILE    *errtpcc;
char    *errFile = "C:/temp/tpcc_encina.out";
enc_status_t enc_status;

#define NewOrder_code    NEWO_TRANS
#define Payment_code     PAYMENT_TRANS
#define OrderStatus_code ORDER_STAT_TRANS
#define Delivery_code    DELIVERY_TRANS
#define StockLevel_code  STOCK_TRANS


#define INT_ENV_VALUE(var, default) \
  (var = getenv(#var) ? atoi(getenv(#var)) : default)


#define PRE_RPC_WORK(headerP, tran, sub_tran) \
        if (iStatsFrequency > 0) \
                        pre_rpc(headerP, tran, sub_tran); \
                else
 \
                        (headerP)->stats = 0;
#define POST_RPC_WORK(headerP, tran) \
        if (iStatsFrequency > 0) \
                        post_rpc(headerP, tran)

/*  CALTPCC
 *  Macro to sends 1 RPC and then handles any errors.
 *
 *  The macro takes the name of the RPC (e.g., NewOrder)
 *  and makes the RPC by calling the appropriate function
 *  (e.g., impTPCCNewOrder).
 */
#define CALLTPCC(name,length,dataP,header,trpcStatusP)            \
{                                                                 \
  UTIL_CONCAT(impTPCC,name)(length,dataP,&header,trpcStatusP);        \
  if (*(trpcStatusP)) {                                           \
    char msg[100];                                                \
    sprintf(msg, "TRPC error during impTPCC%s", UTIL_STRING(name));    \
    header.returncode = TRPC_ERROR;                               \
    encina_error_message(msg, *(trpcStatusP));                    \
  } else if ((header.returncode != TPCC_SUCCESS) &&               \
          (header.returncode != INVALID_NEWO)) {                  \
    char msg[100];                                                \
    sprintf(msg, "App error during impTPCC%s: ", UTIL_STRING(name));   \
    encina_error_message(msg, header.returncode);                 \
  }                                                               \
}

/*
 *  pre_rpc --   For debug purposes
 *
 *  Called before an RPC is made.
 *  Set the state of the thread and keep track of the time the RPC is sent.
 *  This is used by the Background thread to report the state of the client.
 */
static void pre_rpc(data_header *headerP,
                int tran_type,
                int sub_tran_type)
{
   if (iStatsFrequency < 1) {
                        headerP->stats = 0;
   } else {
      int num;
      num = ++ (pClientInfo->tran[tran_type].num);
      headerP->stats = (num % iStatsFrequency==0) ? 1 : 0;
      if (headerP->stats)
                        { /* measure the time for RT */
                                get_local_time(&headerP->clnt_start);
                        headerP->srv_start.sec = 0;   /* initialize the server time */
                        headerP->srv_start.usec = 0;
                        headerP->srv_end.sec = 0;
                        headerP->srv_end.usec = 0;
                        }
   }
}

/*
 *  post_rpc
 *
 *  Called when the RPC returns from the server
 *
 *  Keeps track of the client response time and the server response time
 *  as well as the state of the thread.  This is used by the background
 *  debug thread to report the state of the client
 */
```

```
static void post_rpc(data_header *headerP,
                                int tran_type)
{
   double time_diff;
   int tran_failed;
   struct timeval start_time, end_time;

   if (headerP->stats)
                        get_local_time(&headerP->clnt_end);
   else
                        return;

   /* Store the info for each client.
    * Note: Since we don't use mutex for performance reason, pClientInfo
    *       may not be accurate if more than one thread work on the same
    *       data at a same time. But this should give us reasonable info.
    */
   if ((headerP->returncode == TPCC_SUCCESS) ||
      (headerP->returncode == INVALID_NEWO)) {
     tran_failed = 0;
   } else {
     pClientInfo->tran[tran_type].errs ++;
     pClientInfo->errors ++;
     tran_failed = 1;
   }
   if (headerP->stats && tran_type <= MAX_TRAN_TYPE && tran_type > 0
      && !tran_failed) {
        /* update total server round trip response time */
                start_time.tv_sec = headerP->srv_start.sec;
                start_time.tv_usec = headerP->srv_start.usec;
                end_time.tv_sec = headerP->srv_end.sec;
                end_time.tv_usec = headerP->srv_end.usec;
        time_diff = time_diff_ms(&end_time, &start_time);
        pClientInfo->tran[tran_type].RTtotal[1] += time_diff;
                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 = perfClntDataInit();
   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),statu
s);
      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),st
atus);
   CHK_STATUS(status, MON_INITCLIENT_FAILED,
          "mon_InitClient failed");
   DPRINT(("mon_SecuritySetDefaults-> authn %d, authz %d\n",
                client_authnLevel, client_authzSvc));
   ENCINA_CALL_RC("mon_SecuritySetDefaults",

mon_SecuritySetDefaults(client_authnLevel,client_authzSvc),
          status);
   CHK_STATUS(status, MON_SECURITYSET_FAILED,
          "mon_SecuritySetDefaults failed");

   ENCINA_CALL_RC("mon_SetHandleCacheRefreshInterval",
                     mon_SetHandleCacheRefreshInterval(300), status);
   CHK_STATUS(status, MON_SETREFRESHINTERVAL_FAILED,
          "mon_SetHandleCacheRefreshInterval failed");

   {
      dbInfo_data_t data;
      trpc_status_t trpcStatus;
      /* Get DB Info -- currently id does not do anything
        but it will tell us if there is a server out there.
        Better to know instead of when all the terminals
        are up and ready
       */
      impTPCCNOInfo(&data, &trpcStatus);
      if (trpcStatus) {
         char msg[100];
         sprintf(msg, "TRPC error during db info at init.");
         encina_error_message(msg, trpcStatus);
         CHK_STATUS(33,NOINFO_TRPC_ERROR,
                "TRPC error during db info at init");
      }
   }

   client_enrolled = 1;
   MUTEX_UNLOCK(&init_lock);
   err_printf("end of enroll_client\n");
}


/*-----------------------------------------------------------------*/
/*       Read environment paramaters and registry entries      */
/*-----------------------------------------------------------------*/
static void read_mon_environment()
{
   char *env_str;
   char    *registryKey = "SOFTWARE\\TransarcCorporation\\TxTpcc";
   HKEY    hKey;
   DWORD   size;
   DWORD   type;
   char    szTmp[256];

   cellName = getenv("ENCINA_TPM_CELL");
   CHECK_ENVIRON(cellName, "ENCINA_TPM_CELL");

   if (env_str = getenv("TPCC_ENV_RETRIEVE")) {
      envRetrieval = atoi(env_str);
   }

   if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE, registryKey, 0,
KEY_READ, &hKey) != ERROR_SUCCESS )
      return;

   size = sizeof(szTmp);
   if ( RegQueryValueEx(hKey, "StatsFrequency", 0, &type, szTmp,
&size)==ERROR_SUCCESS)
      iStatsFrequency = atoi(szTmp);

   RegCloseKey(hKey);
}
```

mon_client.h

```
/*
 *              mon_client.h
 *
 */

#ifndef MON_CLIENT_H
#define MON_CLIENT_H

#define MUTEX_T CRITICAL_SECTION
#define MUTEX_LOCK(a) EnterCriticalSection(a)
#define MUTEX_UNLOCK(a) LeaveCriticalSection(a)
#define MUTEX_INIT(mut)   InitializeCriticalSection(mut)
#define MUTEX_DESTROY(mut) DeleteCriticalSection(mut)
#define ERROUT errtpcc

/*initialization status */
#define INIT_SUCCESS 0
#define INIT_FAILED  1
#define CELL_NAME_UNAVAILABLE 2
#define MON_RETRIEVEENABLE_FAILED 3
#define MON_INITCLIENT_FAILED 4
#define MON_SECURITYSET_FAILED 5
#define MON_SETREFRESHINTERVAL_FAILED 6
#define NOINFO_TRPC_ERROR 7
#define ENROLL_CLIENT_EXCEPTION 8
#define ERROUT_FILE_NOT_FOUND 9
#define LOG_FILE_NOT_FOUND 10
#define TPCC_KEY_NOT_FOUND 11
#define TERM_ALLOC_FAILED 12

/*
 *  Routines and declarations that are common to all clients
 */
#if defined(__cplusplus)
extern "C" {
#endif
int send_new_order(long, unsigned char *);
int send_payment(long, unsigned char *);
int send_order_status(long, unsigned char *);
int send_delivery(long, unsigned char *);
int send_stock_level(long, unsigned char *);
void enroll_client();
#if defined(__cplusplus)
}
#endif

#endif  /*  MON_CLIENT_H  */
```

neworder.h

```
#ifndef TRANSARC_neworder_h
#define TRANSARC_neworder_h

#include <trpc/trpc.h>
#include "_neworder.h"

#include <encina/c_prologue.h>

#if defined(BUILDDLL)
#define DLLEXPORT __declspec( dllexport )
#else
#define DLLEXPORT extern
#endif

#ifndef ENCINA_STUB_CALLING
#define ENCINA_STUB_CALLING ENCINA_RPC_CALLING
#endif

#define neworder_v1_0_c_ifspec_neworder_v1_0_c_ifspec
#define neworder_v1_0_s_ifspec _neworder_v1_0_s_ifspec

typedef struct neworder_v1_0_epv {
void (ENCINA_STUB_CALLING *impTPCCNewOrder) (
#ifdef IDL_PROTOTYPES

                        idl_long_int length,
                        idl_char *dataP,
                        data_header *headerP,
                        trpc_status_t *trpcStatus
#endif
);

void (ENCINA_STUB_CALLING *impTPCCNOInfo) (
#ifdef IDL_PROTOTYPES

                        dbInfo_data_t *dataP,
                        trpc_status_t *trpcStatus
#endif
);

} neworder_v1_0_epv_t;

DLLEXPORT void ENCINA_STUB_CALLING impTPCCNewOrder (
#ifdef IDL_PROTOTYPES

                        idl_long_int length,
                        idl_char *dataP,
                        data_header *headerP,
                        trpc_status_t *trpcStatus
#endif
);

DLLEXPORT void ENCINA_STUB_CALLING impTPCCNOInfo (
#ifdef IDL_PROTOTYPES

                        dbInfo_data_t *dataP,
                        trpc_status_t *trpcStatus
#endif
);

trpc_handle_t           ENCINA_CALLING mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
                mon_handle_t        handle,
                trpc_tranInfo_t     *tranInfoP,
                trpc_ifSpec_t *ifSpecP
#endif
                );

void            ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
                mon_handle_t        handle,
                trpc_handle_t       trpcHandle,
                trpc_tranInfo_t     *tranInfoP,
                trpc_ifSpec_t *ifSpecP
#endif
                );

trpc_handle_t           ENCINA_CALLING mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
                mon_handle_t        handle,
                trpc_tranInfo_t     *tranInfoP,
                trpc_ifSpec_t *ifSpecP
#endif
                );

void            ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
```

```
        mon_handle_t        handle,
        trpc_handle_t       trpcHandle,
        trpc_tranInfo_t     *tranInfoP,
        trpc_ifSpec_t *ifSpecP
#endif
        );

extern neworder_v1_0_epv_t      neworder_v1_0_client_epv;
extern _neworder_v1_0_epv_t     neworder_v1_0_manager_epv;
extern rpc_mgr_epv_t            neworder_v1_0_mgr_epv;

#include <encina/c_epilogue.h>
#endif /* TRANSARC_neworder_h */
```

orderstatus.h

```
#ifndef TRANSARC_orderstatus_h
#define TRANSARC_orderstatus_h

#include <trpc/trpc.h>
#include "_orderstatus.h"

#include <encina/c_prologue.h>

#if defined(BUILDDLL)
#define DLLEXPORT __declspec( dllexport )
#else
#define DLLEXPORT extern
#endif

#ifndef ENCINA_STUB_CALLING
#define ENCINA_STUB_CALLING ENCINA_RPC_CALLING
#endif

#define orderstatus_v1_0_c_ifspec       _orderstatus_v1_0_c_ifspec
#define orderstatus_v1_0_s_ifspec       _orderstatus_v1_0_s_ifspec

typedef struct orderstatus_v1_0_epv {
void (ENCINA_STUB_CALLING *impTPCCOrderStatus) (
#ifdef IDL_PROTOTYPES

                idl_long_int length,
                idl_char *dataP,
                data_header *headerP,
                trpc_status_t *trpcStatus
#endif
);

} orderstatus_v1_0_epv_t;

DLLEXPORT void ENCINA_STUB_CALLING impTPCCOrderStatus (
#ifdef IDL_PROTOTYPES

                idl_long_int length,
                idl_char *dataP,
                data_header *headerP,
                trpc_status_t *trpcStatus
#endif
);

trpc_handle_t       ENCINA_CALLING mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
        mon_handle_t        handle,
        trpc_tranInfo_t     *tranInfoP,
        trpc_ifSpec_t *ifSpecP
#endif
        );

void        ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
        mon_handle_t        handle,
        trpc_handle_t       trpcHandle,
        trpc_tranInfo_t     *tranInfoP,
        trpc_ifSpec_t *ifSpecP
#endif
        );

trpc_handle_t       ENCINA_CALLING mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
        mon_handle_t        handle,
        trpc_tranInfo_t     *tranInfoP,
        trpc_ifSpec_t *ifSpecP
#endif
        );

void        ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
        mon_handle_t        handle,
        trpc_handle_t       trpcHandle,
        trpc_tranInfo_t     *tranInfoP,
        trpc_ifSpec_t *ifSpecP
#endif
        );

extern orderstatus_v1_0_epv_t   orderstatus_v1_0_client_epv;
extern _orderstatus_v1_0_epv_t  orderstatus_v1_0_manager_epv;
extern rpc_mgr_epv_t            orderstatus_v1_0_mgr_epv;

#include <encina/c_epilogue.h>
#endif /* TRANSARC_orderstatus_h */
```

payment.h

```
#ifndef TRANSARC_payment_h
#define TRANSARC_payment_h

#include <trpc/trpc.h>
#include "_payment.h"

#include <encina/c_prologue.h>

#if defined(BUILDDLL)
#define DLLEXPORT __declspec( dllexport )
#else
#define DLLEXPORT extern
#endif

#ifndef ENCINA_STUB_CALLING
#define ENCINA_STUB_CALLING ENCINA_RPC_CALLING
#endif

#define payment_v1_0_c_ifspec  _payment_v1_0_c_ifspec
#define payment_v1_0_s_ifspec  _payment_v1_0_s_ifspec

typedef struct payment_v1_0_epv {
void (ENCINA_STUB_CALLING *impTPCCPayment) (
#ifdef IDL_PROTOTYPES

                idl_long_int length,
                idl_char *dataP,
                data_header *headerP,
                trpc_status_t *trpcStatus
#endif
);

} payment_v1_0_epv_t;

DLLEXPORT void ENCINA_STUB_CALLING impTPCCPayment (
```

```
#ifdef IDL_PROTOTYPES

                        idl_long_int length,
                        idl_char *dataP,
                        data_header *headerP,
                        trpc_status_t *trpcStatus
#endif
);

trpc_handle_t           ENCINA_CALLING mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
        mon_handle_t        handle,
        trpc_tranInfo_t     *tranInfoP,
        trpc_ifSpec_t *ifSpecP
#endif
        );

void        ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
        mon_handle_t        handle,
        trpc_handle_t       trpcHandle,
        trpc_tranInfo_t     *tranInfoP,
        trpc_ifSpec_t *ifSpecP
#endif
        );

trpc_handle_t           ENCINA_CALLING mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
        mon_handle_t        handle,
        trpc_tranInfo_t     *tranInfoP,
        trpc_ifSpec_t *ifSpecP
#endif
        );

void        ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
        mon_handle_t        handle,
        trpc_handle_t       trpcHandle,
        trpc_tranInfo_t     *tranInfoP,
        trpc_ifSpec_t *ifSpecP
#endif
        );

extern payment_v1_0_epv_t   payment_v1_0_client_epv;
extern _payment_v1_0_epv_t  payment_v1_0_manager_epv;
extern rpc_mgr_epv_t        payment_v1_0_mgr_epv;

#include <encina/c_epilogue.h>
#endif /* TRANSARC_payment_h */
```

readregistry.cpp

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


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

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

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

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

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

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

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

        pReg->dwNumberOfDeliveryThreads = 0;
        size = sizeof(dwTmp);
```

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

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

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

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

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

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

        RegCloseKey(hKey);

        return FALSE;
}
```

readregistry.h

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

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

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

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

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );
```

resource.h

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

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        129
#define _APS_NEXT_COMMAND_VALUE         40001
```

```
#define _APS_NEXT_CONTROL_VALUE         1024
#define _APS_NEXT_SYMED_VALUE           101
#endif
#endif
```

stocklevel.h

```
#ifndef TRANSARC_stocklevel_h
#define TRANSARC_stocklevel_h

#include <trpc/trpc.h>
#include "_stocklevel.h"

#include <encina/c_prologue.h>

#if defined(BUILDDLL)
#define DLLEXPORT __declspec( dllexport )
#else
#define DLLEXPORT extern
#endif

#ifndef ENCINA_STUB_CALLING
#define ENCINA_STUB_CALLING ENCINA_RPC_CALLING
#endif

#define stocklevel_v1_0_c_ifspec stocklevel_v1_0_c_ifspec
#define stocklevel_v1_0_s_ifspec stocklevel_v1_0_s_ifspec

typedef struct stocklevel_v1_0_epv {
void (ENCINA_STUB_CALLING *impTPCCStockLevel) (
#ifdef IDL_PROTOTYPES

                        idl_long_int length,
                        idl_char *dataP,
                        data_header *headerP,
                        trpc_status_t *trpcStatus
#endif
);

} stocklevel_v1_0_epv_t;

DLLEXPORT void ENCINA_STUB_CALLING impTPCCStockLevel (
#ifdef IDL_PROTOTYPES

                        idl_long_int length,
                        idl_char *dataP,
                        data_header *headerP,
                        trpc_status_t *trpcStatus
#endif
);

trpc_handle_t           ENCINA_CALLING mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
            mon_handle_t        handle,
            trpc_tranInfo_t     *tranInfoP,
            trpc_ifSpec_t *ifSpecP
#endif
            );

void        ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
            mon_handle_t        handle,
            trpc_handle_t       trpcHandle,
            trpc_tranInfo_t     *tranInfoP,
            trpc_ifSpec_t *ifSpecP
#endif
            );

trpc_handle_t           ENCINA_CALLING mon_handle_t_tranBind(
#ifdef IDL_PROTOTYPES
            mon_handle_t        handle,
            trpc_tranInfo_t     *tranInfoP,
            trpc_ifSpec_t *ifSpecP
#endif
            );

void        ENCINA_CALLING mon_handle_t_tranUnBind(
#ifdef IDL_PROTOTYPES
            mon_handle_t        handle,
            trpc_handle_t       trpcHandle,
            trpc_tranInfo_t     *tranInfoP,
            trpc_ifSpec_t *ifSpecP
#endif
            );

extern stocklevel_v1_0_epv_t    stocklevel_v1_0_client_epv;
extern _stocklevel_v1_0_epv_t   stocklevel_v1_0_manager_epv;
extern rpc_mgr_epv_t            stocklevel_v1_0_mgr_epv;

#include <encina/c_epilogue.h>
#endif /* TRANSARC_stocklevel_h */
```

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.
 */

#ifndef _INC_Spinlock

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


/*************************************************************
```

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

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

                #ifdef _DEBUG
                        // Counters for debugging builds.
                        volatile LONG   TotalLocks;
                        volatile LONG   TotalSleeps;
                        volatile LONG   TotalSpins;
                        volatile LONG   TotalWaits;
                #endif

                public:
                        // Public functions.

                        Spinlock( void );

                        inline BOOL ClaimLock( BOOL Wait =
TRUE );
                        inline void ReleaseLock( void );
                        ~Spinlock( void );
                        // Disabled operations.
                        Spinlock( const Spinlock & Copy );
                        void operator=( const Spinlock & Copy );

                private:
                        // Private functions.
                        inline BOOL ClaimSpinlock( volatile LONG
*sl );
                        void WaitForLock( void );
                        void WakeAllSleepers( void );
        };


/************************************************************
        *
        *   A guaranteed atomic exchange.
        *
        *   An attempt is made to claim the Spinlock.  This action is
        *   guaranteed to be atomic.
        *
************************************************************/

        inline BOOL Spinlock::ClaimSpinlock( volatile LONG *Spinlock )
        {
                #ifdef _DEBUG
                        InterlockedIncrement( (LPLONG) &
TotalLocks );
                #endif
                return ( ((*Spinlock) == LockOpen) &&
(InterlockedExchange( (LPLONG)Spinlock, LockClosed ) == LockOpen) );
        }

        /************************************************
        *
        *   Claim the Spinlock.
        *
        *   Claim the lock if available else wait or exit.
        *
        ************************************************/

        inline BOOL Spinlock::ClaimLock( BOOL Wait )
        {
                if ( ! ClaimSpinlock( (volatile LONG*) & m_Spinlock ) )
                {
                        if ( Wait )
                                WaitForLock();
                        return Wait;
                }
                return TRUE;
        }

        /************************************************
        *
        *   Release the Spinlock.
        *
        *   Release the lock and if needed wakeup any sleepers.
        *
        ************************************************/

        inline void Spinlock::ReleaseLock( void )
        {
                m_Spinlock = LockOpen;
                if ( Waiting > 0 )
                        WakeAllSleepers();
        }

        #define _INC_Spinlock

#endif
```

tm_com_dll.dsp

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

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

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

# Begin Project
```

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

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

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

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

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

!ENDIF

# Begin Target

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

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

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

tpcc.cpp

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

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

#include <sqltypes.h>

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

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

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

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

```
#include "..\..\db_odbc_dll\src\tpcc_odbc.h"		// ODBC implementation of TPC-C txns

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

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


#define	LEN_ERR_STRING	256

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

char szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

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

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

static	CRITICAL_SECTION			TermCriticalSection;

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

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

// For deferred Delivery txns:

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

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

// configuration settings from registry
TPCCREGISTRYDATA		Reg;


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

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

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

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

	try
	{
		switch( ul_reason_for_call )
		{
			case DLL_PROCESS_ATTACH:
				{
					DWORD dwSize =
MAX_COMPUTERNAME_LENGTH+1;

GetComputerName(szMyComputerName, &dwSize);

szMyComputerName[dwSize] = 0;
				}


DisableThreadLibraryCalls((HMODULE)hModule);

InitializeCriticalSection(&TermCriticalSection);

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

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

				TermInit();

				// load DLL for txn monitor
```

```
                                        if (Reg.eTxnMon == TUXEDO)
                                        {
                                                strcpy( szDllName,
Reg.szPath );
                                                strcat( szDllName,
"tpcc_tuxedo.dll");
                                                hLibInstanceTm =
LoadLibrary( szDllName );
                                                if (hLibInstanceTm ==
NULL)
                                                        throw new
CWEBCLNT_ERR( ERR_LOADDLL_FAILED, szDllName, GetLastError() );

                                                // get function pointer to
wrapper for class constructor

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

pCTPCC_ENCINA_new = (TYPE_CTPCC_ENCINA*)
GetProcAddress(hLibInstanceTm,"CTPCC_ENCINA_new");

pCTPCC_ENCINA_post_init = (TYPE_CTPCC_ENCINA*)
GetProcAddress(hLibInstanceTm,"CTPCC_ENCINA_post_init");
                                                if
(pCTPCC_ENCINA_new == NULL)
                                                        throw new
CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED, szDllName,
GetLastError() );
                                        }
                                        else if (Reg.eTxnMon == COM)
                                        {
                                                strcpy( szDllName,
Reg.szPath );
                                                strcat( szDllName,
"tpcc_com.dll");
                                                hLibInstanceTm =
LoadLibrary( szDllName );
                                                if (hLibInstanceTm ==
NULL)
                                                        throw new
CWEBCLNT_ERR( ERR_LOADDLL_FAILED, szDllName, GetLastError() );

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

                                        // load DLL for database connection
                                        if ((Reg.eTxnMon == None) ||
(dwNumDeliveryThreads > 0))
                                        {
                                                if (Reg.eDB_Protocol
== DBLIB)
                                                {
                                                        strcpy(
szDllName, Reg.szPath );
                                                        strcat(
szDllName, "tpcc_dblib.dll");

hLibInstanceDb = LoadLibrary( szDllName );
                                                        if
(hLibInstanceDb == NULL)

throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED, szDllName,
GetLastError() );

                                                        // get
function pointer to wrapper for class constructor

pCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_new");
                                                        if
(pCTPCC_DBLIB_new == NULL)

throw new CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED, szDllName,
GetLastError() );
                                                }
                                                else if
(Reg.eDB_Protocol == ODBC)
                                                {
                                                        strcpy(
szDllName, Reg.szPath );
                                                        strcat(
szDllName, "tpcc_odbc.dll");

hLibInstanceDb = LoadLibrary( szDllName );
                                                        if
(hLibInstanceDb == NULL)

throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED, szDllName,
GetLastError() );

                                                        // get
function pointer to wrapper for class constructor

pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
                                                        if
(pCTPCC_ODBC_new == NULL)

throw new CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED, szDllName,
GetLastError() );
                                                }
                                        }

                                        if (dwNumDeliveryThreads)
                                        {
                                                // for deferred delivery
txns:
                                                hDoneEvent =
CreateEvent( NULL, TRUE /* manual reset */, FALSE /* initially not signalled
*/, NULL );
```

```
InitializeCriticalSection(&DelBuffCriticalSection);
                                        hWorkerSemaphore =
CreateSemaphore( NULL, 0, dwDelBuffSize, NULL );
                                        dwDelBuffFreeCount =
dwDelBuffSize;

                                        InitJulianTime(NULL);

                                        // create unique log file
name based on delilog-yymmdd-hhmm.log
                                        SYSTEMTIME Time;
                                        GetLocalTime( &Time
);
                                        wsprintf( szLogFile,
"%sdelivery-%2.2d%2.2d%2.2d-%2.2d%2.2d.log",

Reg.szPath, Time.wYear % 100, Time.wMonth, Time.wDay, Time.wHour,
Time.wMinute );
                                        txnDelilog = new
CTxnLog(szLogFile, TXN_LOG_WRITE);

                                        //write event into txn log
for START

txnDelilog->WriteCtrlRecToLog(TXN_EVENT_START, szMyComputerName,
sizeof(szMyComputerName));

                                        // allocate structures for
delivery buffers and thread mgmt
                                        pDeliHandles = new
HANDLE[dwNumDeliveryThreads];
                                        pDelBuff = new
DELIVERY_TRANSACTION[dwDelBuffSize];
                                        // launch
DeliveryWorkerThread to perform actual delivery txns
                                        for(i=0;
i<dwNumDeliveryThreads; i++)
                                        {

pDeliHandles[i] = (HANDLE) _beginthread( DeliveryWorkerThread, 0, NULL );
                                                if
(pDeliHandles[i] == INVALID_HANDLE_VALUE)

throw new CWEBCLNT_ERR( ERR_DELIVERY_THREAD_FAILED );
                                        }
                                }

                                break;

                        case DLL_PROCESS_DETACH:
                                if (dwNumDeliveryThreads)
                                {
                                        if (txnDelilog != NULL)
                                        {
                                                //write event
into txn log for STOP

txnDelilog->WriteCtrlRecToLog(TXN_EVENT_STOP, szMyComputerName,
sizeof(szMyComputerName));

                                                // This will
do a clean shutdown of the delivery log file
                                                CTxnLog
*txnDelilogLocal = txnDelilog;
                                                txnDelilog=
NULL;
                                                delete
txnDelilogLocal;
                                        }
                                                delete [] pDeliHandles;
                                                delete [] pDelBuff;

                                                CloseHandle(
hWorkerSemaphore );
                                                CloseHandle(
hDoneEvent );

DeleteCriticalSection(&DelBuffCriticalSection);
                                        }


DeleteCriticalSection(&TermCriticalSection);

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

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

                                        Sleep(500);
                                        break;

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

                return TRUE;
}


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

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

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

```
	return TRUE;
}

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

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

	TermDeleteAll();
	return TRUE;
}


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

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

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


#ifdef ICECAP
	StartCAP();
#endif

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

		if (TermId != 0)
		{
			if ( TermId < 0 || TermId >= Term.iNumEntries || Term.pClientData[TermId].iNextFree != -1 )
			{
				// debugging...
				char szTmp[128];
				wsprintf( szTmp, "Invalid term ID; TermId = %d", TermId );
				WriteMessageToEventLog( szTmp );

				throw new CWEBCLNT_ERR( ERR_INVALID_TERMID );
			}

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

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


		switch(iCmd)
		{
		case 0:
			WelcomeForm(pECB, szBuffer);
			break;
		case 1:
			switch( FormId )
			{
				case WELCOME_FORM:
				case MAIN_MENU_FORM:
					break;
				case NEW_ORDER_FORM:
					ProcessNewOrderForm(pECB, TermId, szBuffer);
					break;
				case PAYMENT_FORM:
					ProcessPaymentForm(pECB, TermId, szBuffer);
					break;
				case DELIVERY_FORM:
					ProcessDeliveryForm(pECB, TermId, szBuffer);
					break;
				case ORDER_STATUS_FORM:
					ProcessOrderStatusForm(pECB, TermId, szBuffer);
					break;
				case STOCK_LEVEL_FORM:
					ProcessStockLevelForm(pECB, TermId, szBuffer);
					break;
			}
			break;
		case 2:
			// new-order selected from menu; display new-order input form
			MakeNewOrderForm(TermId, NULL, INPUT_FORM, szBuffer);
			break;
```

```
                        case 3:
                                // payment selected from menu; display
payment input form
                                MakePaymentForm(TermId, NULL,
INPUT_FORM, szBuffer);
                                break;
                        case 4:
                                // delivery selected from menu; display delivery
input form
                                MakeDeliveryForm(TermId, NULL,
INPUT_FORM, szBuffer);
                                break;
                        case 5:
                                // order-status selected from menu; display
order-status input form
                                MakeOrderStatusForm(TermId, NULL,
INPUT_FORM, szBuffer);
                                break;
                        case 6:
                                // stock-level selected from menu; display
stock-level input form
                                MakeStockLevelForm(TermId, NULL,
INPUT_FORM, szBuffer);
                                break;
                        case 7:
                                // ExitCmd
                                TermDelete(TermId);
                                WelcomeForm(pECB, szBuffer);
                                break;
                        case 8:
                                SubmitCmd(pECB, szBuffer);
                                break;
                        case 9:
                                // menu
                                MakeMainMenuForm(TermId,
Term.pClientData[TermId].iSyncId, szBuffer);
                                break;
                        case 10:
                                // CMD=Clear
                                // resets all connections; should only be used
when no other connections are active
                                TermDeleteAll();
                                TermInit();
                                WelcomeForm(pECB, szBuffer);
                                break;
                        case 11:    // CMD=Stats
                                StatsCmd(pECB, szBuffer);
                                break;
                        }
                }
                catch (CBaseErr *e)
                {
                        ErrorForm( pECB, e->ErrorType(), e->ErrorNum(),
TermId, iSyncId, e->ErrorText(), szBuffer );
                        delete e;
                }
                catch (...)
                {
                        ErrorForm( pECB, ERR_TYPE_WEBDLL, 0, TermId,
iSyncId, "Error: Unhandled exception in Web Client.", szBuffer );
                }

#ifdef ICECAP
                StopCAP();
#endif


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

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

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


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

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

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

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

      (VOID) DeregisterEventSource(hEventSource);
   }
}


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

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

                DELIVERY_TRANSACTION                    delivery;
                PDELIVERY_DATA
pDeliveryData;
                TXN_RECORD_TPCC_DELIV_DEF       txnDeliRec;

                DWORD                                                   index;
                HANDLE                                                  handles[2];
```

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

	assert(txnDelilog != 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[] =
        {
                "Process", "..NewOrder..", "..Payment..", "..Delivery..",
"..Order-Status..", "..Stock-Level..",
                "..Exit..", "Submit", "Menu", "Clear", "Stats", ""
        };

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

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

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

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

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

/* FUNCTION: void WelcomeForm
 *
 */

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

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

"<B><BIG>Microsoft TPC-C Web Client (ver 4.20)</BIG></B> <BR> <BR>"
                                                        "<font
face=\"Courier New\"><PRE>"
                                                        "Compiled:
"__DATE__", "__TIME__" <BR>"
                                                        "Source:
"__FILE__" ("__TIMESTAMP__") <BR>"

"</PRE></font>"
                                                        "<FORM
ACTION=\"tpcc.dll\" METHOD=\"GET\">"
                                                        "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"
                                                        "<INPUT
TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
                                                        "<INPUT
TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"1\">"
                                                        "<INPUT
TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"0\">"
                                                        "<INPUT
TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"0\">"
```

```
                                                        "<INPUT
TYPE=\"hidden\" NAME=\"VERSION\" VALUE=\"" WEBCLIENT_VERSION
"\">"
                                );

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

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

        if (Reg.eTxnMon == None)
                // connection options may be specified when not using a txn
monitor
                sprintf( szTmp,         "Please enter your database options
for this connection:<BR>"
                                                        "<font
face=\"Courier New\" color=\"blue\"><PRE>"
                                                        "DB Server
 = <INPUT NAME=\"db_server\" SIZE=20 VALUE=\"%s\"><BR>"
                                                        "DB User ID
  = <INPUT NAME=\"db_user\" SIZE=20 VALUE=\"%s\"><BR>"
                                                        "DB
Password  = <INPUT NAME=\"db_passwd\" SIZE=20 VALUE=\"%s\"><BR>"
                                                        "DB Name
  = <INPUT NAME=\"db_name\" SIZE=20 VALUE=\"%s\"><BR>"

"</PRE></font>"
                                        , Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, Reg.szDbName );
        else
                // if using a txn monitor, connection options are determined
from registry; can't
                // set per user.  show options fyi
                sprintf( szTmp,         "Database options which will be
used by the transaction monitor:<BR>"
                                                        "<font
face=\"Courier New\" color=\"blue\"><PRE>"
                                                        "DB Server
        = <B>%s</B><BR>"
                                                        "DB User ID
         = <B>%s</B><BR>"
                                                        "DB
Password          = <B>%s</B><BR>"
                                                        "DB
Name              = <B>%s</B><BR>"

"</PRE></font>"
                                        , Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, Reg.szDbName );
                strcat( szBuffer, szTmp);

                sprintf( szTmp,         "Please enter your Warehouse and District for
this session:<BR>"
                                                        "<font face=\"Courier
New\" color=\"blue\"><PRE>" );
                strcat( szBuffer, szTmp);
                strcat( szBuffer,       "Warehouse ID = <INPUT NAME=\"w_id\"
SIZE=4><BR>"
                                                        "District ID
= <INPUT NAME=\"d_id\" SIZE=2><BR>"

"</PRE></font><HR>"
                                                        "<INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Submit\">"

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

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

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

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

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

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

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

        // parse district ID
```

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


	iNewTerm = TermAdd();

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

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

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

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

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

	EnterCriticalSection(&TermCriticalSection);

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

	LeaveCriticalSection(&TermCriticalSection);

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

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

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

},
		{	ERR_LOADDLL_FAILED,
			"Load of DLL failed.  DLL="

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

```
                {       ERR_NEWORDER_MISSING_IID_KEY,
        "New Order missing Item Id key \"IID*\"."
                },
                {       ERR_NEWORDER_MISSING_QTY_KEY,
        "New Order Missing Qty key \"Qty##*\"."
                        },
                {
ERR_NEWORDER_MISSING_SUPPW_KEY,
"New Order missing Supp_W key \"SP##*\"."
                },
                {       ERR_NEWORDER_NOITEMS_ENTERED,
                "New Order No order lines entered."
                                                },
                {       ERR_NEWORDER_QTY_INVALID,
                "New Order Qty invalid must be numeric range 1 - 99."
                },
                {       ERR_NEWORDER_QTY_RANGE,
                        "New Order Qty is out of range. Range = 1 to
99."                                    },
                {
ERR_NEWORDER_QTY_WITHOUT_SUPPW,
"New Order Qty field entered without a corresponding Supp_W."
},
                {       ERR_NEWORDER_SUPPW_INVALID,
                "New Order Supp_W invalid data type must be numeric."
                },
                {       ERR_NO_SERVER_SPECIFIED,
                "No Server name specified."

},
                {       ERR_ORDERSTATUS_CID_AND_CLT,
        "Order Status Only Customer ID or Last Name may be entered, not
both."   },
                {       ERR_ORDERSTATUS_CID_INVALID,
        "Order Status Customer ID invalid, range must be numeric 1 - 3000."
},
                {       ERR_ORDERSTATUS_CLT_RANGE,
                "Order Status Customer last name longer than 16
characters."            },
                {       ERR_ORDERSTATUS_DID_INVALID,
        "Order Status District invalid, value must be numeric 1 - 10."
},
                {
ERR_ORDERSTATUS_MISSING_CID_CLT,                "Order
Status Either Customer ID or Last Name must be entered."
},
                {
ERR_ORDERSTATUS_MISSING_CID_KEY,                "Order
Status missing Customer key \"CID*\"."
        },
                {
ERR_ORDERSTATUS_MISSING_CLT_KEY,                "Order
Status missing Customer Last Name key \"CLT*\"."
},
                {
ERR_ORDERSTATUS_MISSING_DID_KEY,                "Order
Status missing District key \"DID*\"."
        },
                {       ERR_PAYMENT_CDI_INVALID,
                "Payment Customer district invalid must be numeric."
                },
                {       ERR_PAYMENT_CID_AND_CLT,
                "Payment Only Customer ID or Last Name may be
entered, not both."      },
                {       ERR_PAYMENT_CUSTOMER_INVALID,
        "Payment Customer data type invalid, must be numeric."
        },
                {       ERR_PAYMENT_CWI_INVALID,
                "Payment Customer Warehouse invalid, must be numeric."
                        },
                {       ERR_PAYMENT_DISTRICT_INVALID,
        "Payment District ID is invalid, must be 1 - 10."
                },
                {       ERR_PAYMENT_HAM_INVALID,
                "Payment Amount invalid data type must be numeric."
                },
                {       ERR_PAYMENT_HAM_RANGE,
                        "Payment Amount out of range, 0 - 9999.99."
                                        },
                {
ERR_PAYMENT_LAST_NAME_TO_LONG,
"Payment Customer last name longer than 16 characters."
},
                {       ERR_PAYMENT_MISSING_CDI_KEY,
        "Payment missing Customer district key \"CDI*\"."
                },
                {       ERR_PAYMENT_MISSING_CID_CLT,
        "Payment Either Customer ID or Last Name must be entered."
        },
                {       ERR_PAYMENT_MISSING_CID_KEY,
        "Payment missing Customer Key \"CID*\"."
                },
                {       ERR_PAYMENT_MISSING_CLT_KEY,
        "Payment missing Customer Last Name key \"CLT*\"."
                },
                {       ERR_PAYMENT_MISSING_CWI_KEY,
        "Payment missing Customer Warehouse key \"CWI*\"."
                },
                {       ERR_PAYMENT_MISSING_DID_KEY,
        "Payment missing District Key \"DID*\"."
                },
                {       ERR_PAYMENT_MISSING_HAM_KEY,
        "Payment missing Amount key \"HAM*\"."
                                },
                {
ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,           "Stock
Level; missing Threshold key \"TT*\"."
        },
                {
ERR_STOCKLEVEL_THRESHOLD_INVALID,               "Stock
Level; Threshold value must be in the range = 1 - 99."          },
                {
ERR_STOCKLEVEL_THRESHOLD_RANGE,
"Stock Level Threshold out of range, range must be 1 - 99."
},
                {       ERR_VERSION_MISMATCH,
                "Invalid version field.  RTE and Web Client are probably
out of sync." },
                {       ERR_W_ID_INVALID,
                        "Invalid Warehouse ID."

        },
                {       0,
                                                ""

                                                                }
        };

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

```
                }
                i++;
        }

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

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

}

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

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

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

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

        *pQueryString = ptr;
        return;

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

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

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

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

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

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

```
	*pQueryString = ptr;
	return atoi(ptr0);

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

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

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

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

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

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

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

	LeaveCriticalSection(&TermCriticalSection);
}

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

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

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

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

	LeaveCriticalSection(&TermCriticalSection);
}

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

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

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

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

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

```
	LeaveCriticalSection(&TermCriticalSection);
	return iNewTerm;
}

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

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

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

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

		LeaveCriticalSection(&TermCriticalSection);
	}
}


/* FUNCTION: MakeErrorForm
 */

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

/* FUNCTION: MakeMainMenuForm
 */

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

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

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

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

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

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

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

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

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

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

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

		strcpy( szForm+c,
			"District: <INPUT NAME=\"DID*\" SIZE=1>
	Date:<BR>"
			"Customer:  <INPUT NAME=\"CID*\"
SIZE=4>   Name:              Credit:     %Disc:<BR>"
			"Order Number:          Number of Lines:
W_tax:       D_tax:<BR> <BR>"
			" Supp_W  Item_Id  Item Name              Qty
Stock  B/G  Price    Amount<BR>"
			"  <INPUT NAME=\"SP00*\" SIZE=4>
<INPUT NAME=\"IID00*\" SIZE=6>                    <INPUT
NAME=\"Qty00*\" SIZE=1><BR>"
			"  <INPUT NAME=\"SP01*\" SIZE=4>
<INPUT NAME=\"IID01*\" SIZE=6>                    <INPUT
NAME=\"Qty01*\" SIZE=1><BR>"
			"  <INPUT NAME=\"SP02*\" SIZE=4>
<INPUT NAME=\"IID02*\" SIZE=6>                    <INPUT
NAME=\"Qty02*\" SIZE=1><BR>"
			"  <INPUT NAME=\"SP03*\" SIZE=4>
<INPUT NAME=\"IID03*\" SIZE=6>                    <INPUT
NAME=\"Qty03*\" SIZE=1><BR>"
			"  <INPUT NAME=\"SP04*\" SIZE=4>
<INPUT NAME=\"IID04*\" SIZE=6>                    <INPUT
NAME=\"Qty04*\" SIZE=1><BR>"
			"  <INPUT NAME=\"SP05*\" SIZE=4>
<INPUT NAME=\"IID05*\" SIZE=6>                    <INPUT
NAME=\"Qty05*\" SIZE=1><BR>"
			"  <INPUT NAME=\"SP06*\" SIZE=4>
<INPUT NAME=\"IID06*\" SIZE=6>                    <INPUT
NAME=\"Qty06*\" SIZE=1><BR>"
			"  <INPUT NAME=\"SP07*\" SIZE=4>
<INPUT NAME=\"IID07*\" SIZE=6>                    <INPUT
NAME=\"Qty07*\" SIZE=1><BR>"
			"  <INPUT NAME=\"SP08*\" SIZE=4>
<INPUT NAME=\"IID08*\" SIZE=6>                    <INPUT
NAME=\"Qty08*\" SIZE=1><BR>"
			"  <INPUT NAME=\"SP09*\" SIZE=4>
<INPUT NAME=\"IID09*\" SIZE=6>                    <INPUT
NAME=\"Qty09*\" SIZE=1><BR>"
			"  <INPUT NAME=\"SP10*\" SIZE=4>
<INPUT NAME=\"IID10*\" SIZE=6>                    <INPUT
NAME=\"Qty10*\" SIZE=1><BR>"
			"  <INPUT NAME=\"SP11*\" SIZE=4>
<INPUT NAME=\"IID11*\" SIZE=6>                    <INPUT
NAME=\"Qty11*\" SIZE=1><BR>"
			"  <INPUT NAME=\"SP12*\" SIZE=4>
<INPUT NAME=\"IID12*\" SIZE=6>                    <INPUT
NAME=\"Qty12*\" SIZE=1><BR>"
			"  <INPUT NAME=\"SP13*\" SIZE=4>
<INPUT NAME=\"IID13*\" SIZE=6>                    <INPUT
NAME=\"Qty13*\" SIZE=1><BR>"
```

```
                                        "  <INPUT NAME=\"SP14*\" SIZE=4>
<INPUT NAME=\"IID14*\" SIZE=6>                  <INPUT
NAME=\"Qty14*\" SIZE=1><BR>"
                                        "Execution Status:
Total:<BR>"
                                        "</font></PRE><HR>"
                                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Process\">"
                                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Menu\">"
                                        "</FORM></HTML>"
                                        );
                }
                else
                {
                        c += wsprintf(szForm+c, "Warehouse: %4.4d   District:
%2.2d                   Date: ",
                                        pNewOrderData->w_id,
                                        pNewOrderData->d_id);

                        if ( bValid )
                        {
                                c += wsprintf(szForm+c, "%2.2d-%2.2d-%4.4d
%2.2d:%2.2d:%2.2d",
                                                pNewOrderData->o_entry_d.day,

pNewOrderData->o_entry_d.month,
                                                pNewOrderData->o_entry_d.year,
                                                pNewOrderData->o_entry_d.hour,

pNewOrderData->o_entry_d.minute,

pNewOrderData->o_entry_d.second);
                        }

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

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

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

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

pNewOrderData->OL[i].ol_supply_w_id,

pNewOrderData->OL[i].ol_i_id,

pNewOrderData->OL[i].ol_i_name,

pNewOrderData->OL[i].ol_quantity,

pNewOrderData->OL[i].ol_stock,

pNewOrderData->OL[i].ol_brand_generic,

pNewOrderData->OL[i].ol_i_price,

pNewOrderData->OL[i].ol_amount );
                                }
                        }
                        else
                        {
                                c += wsprintf(szForm+c,
                                        "%Disc:<BR>"
                                        "Order Number: %8.8d  Number of
Lines:          W_tax:      D_tax:<BR> <BR>"
                                        " Supp_W  Item_Id  Item Name
      Qty  Stock  B/G  Price    Amount<BR>"
                                        , pNewOrderData->o_id);

                                i = 0;
                        }

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

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

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

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

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

        c = wsprintf(szForm,
                "<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD><BODY>"
                "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
                "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">"
```

```
                        "<INPUT TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\" NAME=\"TERMID\"
VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\"
VALUE=\"%d\">"
                        "<PRE><font face=\"Courier\">
Payment<BR>"
                        "Date: "
                        , PAYMENT_FORM, iTermId,
Term.pClientData[iTermId].iSyncId);

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

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

                                , Term.pClientData[iTermId].w_id,
pPaymentData->d_id
                                , pPaymentData->w_street_1,
pPaymentData->d_street_1
                                , pPaymentData->w_street_2,
pPaymentData->d_street_2
                                , pPaymentData->w_city,
pPaymentData->w_state, pPaymentData->w_zip, pPaymentData->w_zip+5
                                , pPaymentData->d_city,
pPaymentData->d_state, pPaymentData->d_zip, pPaymentData->d_zip+5
                                , pPaymentData->c_id,
pPaymentData->c_w_id,          pPaymentData->c_d_id
                                , pPaymentData->c_first,
pPaymentData->c_middle, pPaymentData->c_last
                                , pPaymentData->c_since.day,
pPaymentData->c_since.month,  pPaymentData->c_since.year
                                , pPaymentData->c_street_1,
pPaymentData->c_credit
                                );

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

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

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

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

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

"</BODY></FORM></HTML>");
        }
```

```
}

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

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

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

	if ( bInput )
	{
		strcpy(szForm+c,
			"District: <INPUT NAME=\"DID*\" SIZE=1><BR>"
			"Customer: <INPUT NAME=\"CID*\" SIZE=4>   Name:  <INPUT NAME=\"CLT*\" SIZE=23><BR>"
			"Cust-Balance:<BR> <BR>"
			"Order-Number:  Entry-Date:  Carrier-Number:<BR>"
			"Supply-W  Item-Id  Qty  Amount  Delivery-Date<BR> <BR> <BR> <BR> <BR>"
			" <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR></font></PRE>"
			"<HR><INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\"><INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
			"</BODY></FORM></HTML>" );
	}
	else
	{
		c += wsprintf(szForm+c,
			"District: %2.2d<BR>"
			"Customer: %4.4d   Name: %-16s %-2s %-16s<BR>",
			pOrderStatusData->d_id, pOrderStatusData->c_id,
			pOrderStatusData->c_first, pOrderStatusData->c_middle, pOrderStatusData->c_last);

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

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

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

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

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


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

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

	c = wsprintf(szForm,
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
 *                                              int
                        iTermId     client browser terminal id
 *
 */

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

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

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

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

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

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

        PDELIVERY_DATA pDelivery;

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

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

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

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

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

        PSTOCK_LEVEL_DATA       pStockLevel;

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

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

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

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

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

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

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

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

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

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

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

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

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

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

	pNewOrderData->o_ol_cnt = items;
}

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

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

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

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

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

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

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

```
                                throw new CWEBCLNT_ERR(
ERR_PAYMENT_CID_AND_CLT );
        }

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

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

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

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

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

}

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

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

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

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

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

        if ( *ptr == 0 )
                return FALSE;

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

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

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

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

tpcc.def

```
LIBRARY TPCC.DLL

EXPORTS

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

tpcc.h

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

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

#define TP_MAX_RETRIES
        50


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

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


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

        CTPCC_BASE      *pTxn;

} CLIENTDATA, *PCLIENTDATA;


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

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


        enum WEBERROR
        {
                NO_ERR,
                ERR_COMMAND_UNDEFINED,
                ERR_D_ID_INVALID,
                ERR_DELIVERY_CARRIER_ID_RANGE,
                ERR_DELIVERY_CARRIER_INVALID,
                ERR_DELIVERY_MISSING_OCD_KEY,
                ERR_DELIVERY_THREAD_FAILED,
                ERR_GETPROCADDR_FAILED,
                ERR_HTML_ILL_FORMED,
                ERR_INVALID_SYNC_CONNECTION,
                ERR_INVALID_TERMID,
                ERR_LOADDLL_FAILED,
                ERR_MAX_CONNECTIONS_EXCEEDED,
                ERR_MEM_ALLOC_FAILED,
                ERR_MISSING_REGISTRY_ENTRIES,
                ERR_NEWORDER_CUSTOMER_INVALID,
                ERR_NEWORDER_CUSTOMER_KEY,
                ERR_NEWORDER_DISTRICT_INVALID,
                ERR_NEWORDER_FORM_MISSING_DID,
                ERR_NEWORDER_ITEMID_INVALID,
                ERR_NEWORDER_ITEMID_RANGE,
                ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
                ERR_NEWORDER_MISSING_IID_KEY,
                ERR_NEWORDER_MISSING_QTY_KEY,
                ERR_NEWORDER_MISSING_SUPPW_KEY,
                ERR_NEWORDER_NOITEMS_ENTERED,
                ERR_NEWORDER_QTY_INVALID,
                ERR_NEWORDER_QTY_RANGE,
                ERR_NEWORDER_QTY_WITHOUT_SUPPW,
                ERR_NEWORDER_SUPPW_INVALID,
                ERR_NO_SERVER_SPECIFIED,
                ERR_ORDERSTATUS_CID_AND_CLT,
                ERR_ORDERSTATUS_CID_INVALID,
                ERR_ORDERSTATUS_CLT_RANGE,
                ERR_ORDERSTATUS_DID_INVALID,
                ERR_ORDERSTATUS_MISSING_CID_CLT,
                ERR_ORDERSTATUS_MISSING_CID_KEY,
                ERR_ORDERSTATUS_MISSING_CLT_KEY,
                ERR_ORDERSTATUS_MISSING_DID_KEY,
                ERR_PAYMENT_CDI_INVALID,
                ERR_PAYMENT_CID_AND_CLT,
```

```
                ERR_PAYMENT_CUSTOMER_INVALID,
                ERR_PAYMENT_CWI_INVALID,
                ERR_PAYMENT_DISTRICT_INVALID,
                ERR_PAYMENT_HAM_INVALID,
                ERR_PAYMENT_HAM_RANGE,
                ERR_PAYMENT_LAST_NAME_TO_LONG,
                ERR_PAYMENT_MISSING_CDI_KEY,
                ERR_PAYMENT_MISSING_CID_CLT,
                ERR_PAYMENT_MISSING_CID_KEY,
                ERR_PAYMENT_MISSING_CLT,
                ERR_PAYMENT_MISSING_CLT_KEY,
                ERR_PAYMENT_MISSING_CWI_KEY,
                ERR_PAYMENT_MISSING_DID_KEY,
                ERR_PAYMENT_MISSING_HAM_KEY,
                ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
                ERR_STOCKLEVEL_THRESHOLD_INVALID,
                ERR_STOCKLEVEL_THRESHOLD_RANGE,
                ERR_VERSION_MISMATCH,
                ERR_W_ID_INVALID
        };

class CWEBCLNT_ERR : public CBaseErr
{
        public:

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

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

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

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

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

};

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

//function prototypes

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

tpcc.rc

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

#define APSTUDIO_READONLY_SYMBOLS
////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 2 resource.
//
```

```
#include "afxres.h"

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

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

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

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

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

#endif    // !_MAC


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

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

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

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

#endif    // APSTUDIO_INVOKED


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

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


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

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
    IDD_DIALOG1, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 179
        TOPMARGIN, 7
        BOTTOMMARGIN, 88
    END
END
#endif    // APSTUDIO_INVOKED

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



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


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

tpcc_com.cpp

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

```
 *		PURPOSE:		Source file for TPC-C COM+ class implementation.
 *		Contact:	Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *			4.20.000 - first version
 */

// needed for CoinitializeEx
#define _WIN32_WINNT 0x0400

#include <windows.h>

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

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

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

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

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

	m_bSinglePool = bSinglePool;

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

	m_pTxn = (COM_DATA*)CoTaskMemAlloc(sizeof(COM_DATA));
	if (!m_pTxn)
		throw new CCOMERR( E_FAIL );

	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)
		CoTaskMemFree(m_pTxn);

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

void CTPCC_COM::NewOrder()
{
	int iSize = sizeof(COM_DATA);

	HRESULT hr = m_pNewOrder->NewOrder(&iSize, (unsigned char**)&m_pTxn);
	if (FAILED(hr))
		throw new CCOMERR( hr );

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

```
void CTPCC_COM::Payment()
{
	int iSize = sizeof(COM_DATA);

	HRESULT hr = m_pPayment->Payment(&iSize, (unsigned
char**)&m_pTxn);
	if (FAILED(hr))
		throw new CCOMERR( hr );

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

void CTPCC_COM::StockLevel()
{
	int iSize = sizeof(COM_DATA);

	HRESULT hr = m_pStockLevel->StockLevel(&iSize, (unsigned
char**)&m_pTxn);
	if (FAILED(hr))
		throw new CCOMERR( hr );

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

void CTPCC_COM::OrderStatus()
{
	int iSize = sizeof(COM_DATA);

	HRESULT hr = m_pOrderStatus->OrderStatus(&iSize, (unsigned
char**)&m_pTxn);
	if (FAILED(hr))
		throw new CCOMERR( hr );

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

tpcc_com.h

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

#pragma once

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

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

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

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

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

		int		m_hr;
		int		m_iErrorType;
		int		m_iError;

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

		int ErrorNum() {return m_hr;}

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

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

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

		struct COM_DATA
		{
			int ErrorType;
			int error;
			union
			{
				NEW_ORDER_DATA
NewOrder;
```

```
                                        PAYMENT_DATA
Payment;
                                        DELIVERY_DATA
Delivery;
                                        STOCK_LEVEL_DATA
StockLevel;
                                        ORDER_STATUS_DATA
OrderStatus;
                                } u;
                        } *m_pTxn;

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

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

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


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


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

typedef CTPCC_COM* (TYPE_CTPCC_COM)(BOOL);
```

tpcc_com_all.cpp

```
/*      FILE:           TPCC_COM_ALL.CPP
 *                              Microsoft TPC-C Kit Ver. 4.20.000
 *                              Copyright Microsoft, 1999
 *      All Rights Reserved
 *
 *                              Version 4.10.000 audited by
Richard Gimarc, Performance Metrics, 3/17/99
 *
 *      PURPOSE:        Implementation for TPC-C Tuxedo class.
 *      Contact:  Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *              4.20.000 - updated rev number to match kit
 */

#define STRICT
#define _WIN32_WINNT 0x0400
#define _ATL_APARTMENT_THREADED

#include <stdio.h>
#include <atlbase.h>
//You may derive a class from CComModule and use it if you want to override
//something, but do not change the name of _Module
extern CComModule _Module;

#include <atlcom.h>
#include <initguid.h>
#include <transact.h>
#include <atlimpl.cpp>
#include <comsvcs.h>

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

#include "tpcc_com_ps.h"
#include "..\..\common\src\trans.h"
//tpckit transaction header contains definations of structures specific to TPC-C
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\error.h"
#include "..\..\common\src\ReadRegistry.h"
#include "..\..\db_dblib_dll\src\tpcc_dblib.h"          // DBLIB
implementation of TPC-C txns
#include "..\..\db_odbc_dll\src\tpcc_odbc.h"            // ODBC
implementation of TPC-C txns

#include "resource.h"
#include "tpcc_com_all.h"
#include "tpcc_com_all_i.c"
#include "Methods.h"
#include "..\..\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "..\..\common\src\ReadRegistry.cpp"


CComModule _Module;

BEGIN_OBJECT_MAP(ObjectMap)
        OBJECT_ENTRY(CLSID_TPCC, CTPCC)
        OBJECT_ENTRY(CLSID_NewOrder, CNewOrder)
        OBJECT_ENTRY(CLSID_OrderStatus, COrderStatus)
        OBJECT_ENTRY(CLSID_Payment, CPayment)
        OBJECT_ENTRY(CLSID_StockLevel, CStockLevel)
END_OBJECT_MAP()


// configuration settings from registry
TPCCREGISTRYDATA        Reg;
char
szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

static  HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB        *pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC                 *pCTPCC_ODBC_new;


////////////////////////////////////////////////////////////////////////////
// DLL Entry Point

extern "C"
BOOL WINAPI DllMain(HINSTANCE hInstance, DWORD dwReason,
LPVOID /*lpReserved*/)
{
        char szDllName[128];

        try
        {
```

```
                if (dwReason == DLL_PROCESS_ATTACH)
                {
                        _Module.Init(ObjectMap, hInstance);
                        DisableThreadLibraryCalls(hInstance);

                        DWORD dwSize =
MAX_COMPUTERNAME_LENGTH+1;
                        GetComputerName(szMyComputerName,
&dwSize);
                        szMyComputerName[dwSize] = 0;

                        if ( ReadTPCCRegistrySettings( &Reg ) )
                                throw new CCOMPONENT_ERR(
ERR_MISSING_REGISTRY_ENTRIES );

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

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

                                // get function pointer to wrapper for
class constructor
                                pCTPCC_ODBC_new =
(TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
                                if (pCTPCC_ODBC_new ==
NULL)
                                        throw new
CCOMPONENT_ERR( ERR_GETPROCADDR_FAILED, szDllName,
GetLastError() );
                        }
                        else
                                throw new CCOMPONENT_ERR(
ERR_UNKNOWN_DB_PROTOCOL );
                }
                else if (dwReason == DLL_PROCESS_DETACH)
                        _Module.Term();

        }
        catch (CBaseErr *e)
        {
                WriteMessageToEventLog(e->ErrorText());
                delete e;
                return FALSE;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception
in object DllMain"));
                return FALSE;
        }

        return TRUE;            // OK
}

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

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

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

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

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

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

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

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


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

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

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

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

```
        (LPCTSTR *)lpszStrings,    // array of error strings
        NULL);              // no raw data

      (VOID) DeregisterEventSource(hEventSource);
   }
}

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

/* FUNCTION: CCOMPONENT_ERR::ErrorText
 *
 */

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

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

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

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


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

CTPCC_Common::~CTPCC_Common()
{
        if (m_pTxn)
                delete m_pTxn;
}

HRESULT CTPCC_Common::CallSetComplete()
{
        IObjectContext* pObjectContext = NULL;

        // get our object context
        HRESULT hr = CoGetObjectContext( IID_IObjectContext, (void
**)&pObjectContext );
        pObjectContext->SetComplete();
        ReleaseInterface(pObjectContext);
        return hr;
}

//
//      called by the ctor activator
//
STDMETHODIMP CTPCC_Common::Construct(IDispatch * pUnk)
{
        // Code to access construction string, if needed later...
        //      if (!pUnk)
        //              return E_UNEXPECTED;
        //      IObjectConstructString * pString = NULL;
        //      HRESULT hr =
pUnk->QueryInterface(IID_IObjectConstructString, (void **)&pString);
        //      pString->Release();

        try
        {
                if (Reg.eDB_Protocol == ODBC)
                        m_pTxn = pCTPCC_ODBC_new(
Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, szMyComputerName,
Reg.szDbName );
                else if (Reg.eDB_Protocol == DBLIB)
                        m_pTxn = pCTPCC_DBLIB_new(
Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, szMyComputerName,
Reg.szDbName );
        }
        catch (CBaseErr *e)
        {
                WriteMessageToEventLog(e->ErrorText());
                delete e;
                return E_FAIL;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception
in object ::Construct"));
                return E_FAIL;
        }

        return S_OK;
}

HRESULT CTPCC_Common::NewOrder(int* iSize, UCHAR **txn)
{
        PNEW_ORDER_DATA         pNewOrder;
        COM_DATA                *pData;

        try
        {
                pData = (COM_DATA*)*txn;
                pNewOrder = m_pTxn->BuffAddr_NewOrder();
```

```
                        memcpy(pNewOrder, &pData->u.NewOrder,
sizeof(NEW_ORDER_DATA));
                        m_pTxn->NewOrder();
                        memcpy( &pData->u.NewOrder, pNewOrder,
sizeof(NEW_ORDER_DATA));

                        pData->retval = ERR_SUCCESS;
                        pData->error = 0;
                        return S_OK;
                }
                catch (CBaseErr *e)
                {
                        // check for lost database connection; if yes, component is
toast
                        if ( ((e->ErrorType() == ERR_TYPE_DBLIB) &&
(e->ErrorNum() == 10005)) ||
                                        ((e->ErrorType() == ERR_TYPE_ODBC)
&& (e->ErrorNum() == 10054)) )
                                m_bCanBePooled = FALSE;

                        pData->retval = e->ErrorType();
                        pData->error = e->ErrorNum();
                        delete e;
                        return E_FAIL;
                }
                catch (...)
                {
                        WriteMessageToEventLog(TEXT("Unhandled
exception."));
                        pData->retval = ERR_TYPE_LOGIC;
                        pData->error = 0;
                        m_bCanBePooled = FALSE;
                        return E_FAIL;
                }
}

HRESULT CTPCC_Common::Payment(int* iSize, UCHAR** txn)
{
                PPAYMENT_DATA  pPayment;
                COM_DATA                        *pData;

                try
                {
                        pData = (COM_DATA*)*txn;
                        pPayment = m_pTxn->BuffAddr_Payment();

                        memcpy(pPayment, &pData->u.Payment,
sizeof(PAYMENT_DATA) );
                        m_pTxn->Payment();
                        memcpy( &pData->u.Payment, pPayment,
sizeof(PAYMENT_DATA) );

                        pData->retval = ERR_SUCCESS;
                        pData->error = 0;
                        return S_OK;
                }
                catch (CBaseErr *e)
                {
                        // check for lost database connection; if yes, component is
toast
                        if ( ((e->ErrorType() == ERR_TYPE_DBLIB) &&
(e->ErrorNum() == 10005)) ||
                                        ((e->ErrorType() == ERR_TYPE_ODBC)
&& (e->ErrorNum() == 10054)) )
                                m_bCanBePooled = FALSE;

                        pData->retval = e->ErrorType();
                        pData->error = e->ErrorNum();
                        delete e;
                        return E_FAIL;
                }
                catch (...)
                {
                        WriteMessageToEventLog(TEXT("Unhandled
exception."));
                        pData->retval = ERR_TYPE_LOGIC;
                        pData->error = 0;
                        m_bCanBePooled = FALSE;
                        return E_FAIL;
                }
}

HRESULT CTPCC_Common::StockLevel(int* iSize, UCHAR** txn)
{
                PSTOCK_LEVEL_DATA       pStockLevel;
                COM_DATA                        *pData;

                try
                {
                        pData = (COM_DATA*)*txn;
                        pStockLevel = m_pTxn->BuffAddr_StockLevel();

                        memcpy(pStockLevel, &pData->u.StockLevel,
sizeof(STOCK_LEVEL_DATA) );
                        m_pTxn->StockLevel();
                        memcpy( &pData->u.StockLevel, pStockLevel,
sizeof(STOCK_LEVEL_DATA) );

                        pData->retval = ERR_SUCCESS;
                        pData->error = 0;
                        return S_OK;
                }
                catch (CBaseErr *e)
                {
                        // check for lost database connection; if yes, component is
toast
                        if ( ((e->ErrorType() == ERR_TYPE_DBLIB) &&
(e->ErrorNum() == 10005)) ||
                                        ((e->ErrorType() == ERR_TYPE_ODBC)
&& (e->ErrorNum() == 10054)) )
                                m_bCanBePooled = FALSE;

                        pData->retval = e->ErrorType();
                        pData->error = e->ErrorNum();
                        delete e;
                        return E_FAIL;
                }
                catch (...)
                {
                        WriteMessageToEventLog(TEXT("Unhandled
exception."));
                        pData->retval = ERR_TYPE_LOGIC;
                        pData->error = 0;
                        m_bCanBePooled = FALSE;
                        return E_FAIL;
                }
}

HRESULT CTPCC_Common::OrderStatus(int* iSize, UCHAR** txn)
{
                PORDER_STATUS_DATA      pOrderStatus;
                COM_DATA                        *pData;

                try
                {
                        pData = (COM_DATA*)*txn;
                        pOrderStatus = m_pTxn->BuffAddr_OrderStatus();

                        memcpy(pOrderStatus, &pData->u.OrderStatus,
sizeof(ORDER_STATUS_DATA) );
                        m_pTxn->OrderStatus();
```

```
                        memcpy( &pData->u.OrderStatus, pOrderStatus,
sizeof(ORDER_STATUS_DATA) );

                        pData->retval = ERR_SUCCESS;
                        pData->error = 0;
                        return S_OK;
                }
                catch (CBaseErr *e)
                {
                        // check for lost database connection; if yes, component is
toast
                        if ( ((e->ErrorType() == ERR_TYPE_DBLIB) &&
(e->ErrorNum() == 10005)) ||
                                ((e->ErrorType() == ERR_TYPE_ODBC)
&& (e->ErrorNum() == 10054)) )
                                m_bCanBePooled = FALSE;

                        pData->retval = e->ErrorType();
                        pData->error = e->ErrorNum();
                        delete e;
                        return E_FAIL;
                }
                catch (...)
                {
                        WriteMessageToEventLog(TEXT("Unhandled
exception."));
                        pData->retval = ERR_TYPE_LOGIC;
                        pData->error = 0;
                        m_bCanBePooled = FALSE;
                        return E_FAIL;
                }
}
```

tpcc_com_all.def

```
; tpcc_com_all.def : Declares the module parameters.

LIBRARY      "tpcc_com_all.dll"

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

tpcc_com_all.dsp

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

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

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

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

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

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

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D
"_DEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D
"_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
```

```
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib
odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ..\db_dblib_dll\bin\tpcc_dblib.lib
..\db_odbc_dll\bin\tpcc_odbc.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib
odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept

!ENDIF

# Begin Target

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

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

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

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

SOURCE=.\src\tpcc_com_all.idl

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

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

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

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

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

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

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

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

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

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

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

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

!ENDIF

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

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

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

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

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

tpcc_com_all.h

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

/* this ALWAYS GENERATED file contains the definitions for the interfaces */


 /* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jan 24 20:00:20 2000
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


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

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

#ifndef __tpcc_com_all_h__
#define __tpcc_com_all_h__

/* Forward Declarations */

#ifndef __TPCC_FWD_DEFINED__
#define __TPCC_FWD_DEFINED__

#ifdef __cplusplus
typedef class TPCC TPCC;
#else
typedef struct TPCC TPCC;
#endif /* __cplusplus */

#endif      /* __TPCC_FWD_DEFINED__ */
```

```
#ifndef __NewOrder_FWD_DEFINED__
#define __NewOrder_FWD_DEFINED__

#ifdef __cplusplus
typedef class NewOrder NewOrder;
#else
typedef struct NewOrder NewOrder;
#endif /* __cplusplus */

#endif      /* __NewOrder_FWD_DEFINED__ */


#ifndef __OrderStatus_FWD_DEFINED__
#define __OrderStatus_FWD_DEFINED__

#ifdef __cplusplus
typedef class OrderStatus OrderStatus;
#else
typedef struct OrderStatus OrderStatus;
#endif /* __cplusplus */

#endif      /* __OrderStatus_FWD_DEFINED__ */


#ifndef __Payment_FWD_DEFINED__
#define __Payment_FWD_DEFINED__

#ifdef __cplusplus
typedef class Payment Payment;
#else
typedef struct Payment Payment;
#endif /* __cplusplus */

#endif      /* __Payment_FWD_DEFINED__ */


#ifndef __StockLevel_FWD_DEFINED__
#define __StockLevel_FWD_DEFINED__

#ifdef __cplusplus
typedef class StockLevel StockLevel;
#else
typedef struct StockLevel StockLevel;
#endif /* __cplusplus */

#endif      /* __StockLevel_FWD_DEFINED__ */


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

#ifdef __cplusplus
extern "C"{
#endif

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

/* interface __MIDL_itf_tpcc_com_all_0000 */
/* [local] */

extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_s_ifspec;


#ifndef __TPCCLib_LIBRARY_DEFINED__
#define __TPCCLib_LIBRARY_DEFINED__

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


EXTERN_C const IID LIBID_TPCCLib;

EXTERN_C const CLSID CLSID_TPCC;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_NewOrder;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_OrderStatus;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_Payment;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_StockLevel;

#ifdef __cplusplus

class DECLSPEC_UUID("2668369E-A50D-11D2-BA4E-00C04FBFE08B")
StockLevel;
#endif
#endif /* __TPCCLib_LIBRARY_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

tpcc_com_all.idl

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

```
 *
 *  Change history:
 *                      4.20.000 - first version
 */

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


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

[
        uuid(122A3117-2520-11D3-BA71-00C04FBFE08B),
        version(1.0),
        helpstring("TPC-C 1.0 Type Library")
]
library TPCCLib
{
        importlib("stdole32.tlb");
        importlib("stdole2.tlb");

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


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


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


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


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

};
```

tpcc_com_all.rc

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

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

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

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

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

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

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

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

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

#endif    // APSTUDIO_INVOKED


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

VS_VERSION_INFO VERSIONINFO
 FILEVERSION 1,0,0,1
 PRODUCTVERSION 1,0,0,1
 FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
 FILEFLAGS 0x1L
```

```
#else
 FILEFLAGS 0x0L
#endif
 FILEOS 0x4L
 FILETYPE 0x2L
 FILESUBTYPE 0x0L
BEGIN
   BLOCK "StringFileInfo"
   BEGIN
      BLOCK "040904B0"
      BEGIN
         VALUE "CompanyName", "\0"
         VALUE "FileDescription", "tpcc_com_all Module\0"
         VALUE "FileVersion", "1, 0, 0, 1\0"
         VALUE "InternalName", "TPCCNEWORDER\0"
         VALUE "LegalCopyright", "Copyright 1997\0"
         VALUE "OriginalFilename", "tpcc_com_all.DLL\0"
         VALUE "ProductName", "tpcc_com_all Module\0"
         VALUE "ProductVersion", "1, 0, 0, 1\0"
         VALUE "OLESelfRegister", "\0"
      END
   END
   BLOCK "VarFileInfo"
   BEGIN
      VALUE "Translation", 0x409, 1200
   END
END

#endif    // !_MAC


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

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

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

STRINGTABLE DISCARDABLE
BEGIN
   IDS_PROJNAME          "tpcc_com_all"
END

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



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

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

tpcc_com_all_resource.h

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

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

tpcc_type.h

```
/* Generated by IDL compiler version DEC DCE V2.0.0-6 */
#ifndef tpcc_types_v1_0_included
#define tpcc_types_v1_0_included
#ifndef IDLBASE_H
#include <dce\idlbase.h>
#endif

#ifdef __cplusplus
    extern "C" {
#endif

#ifndef nbase_v0_0_included
#include "dce\nbase.h"
#endif
#define NAME_LENGTH (32)
#define NEWO_INTERFACE (1)
#define PAYMENT_INTERFACE (2)
#define ORDER_STAT_INTERFACE (4)
#define DELIVERY_INTERFACE (8)
#define STOCK_INTERFACE (16)
#define ONLINE_INTERFACES (23)
#define ALL_INTERFACE (65535)
#define NEWO_TRANS (1)
#define PAYMENT_TRANS (2)
#define ORDER_STAT_TRANS (3)
#define DELIVERY_TRANS (4)
#define STOCK_TRANS (5)
#define MAX_TRAN_TYPE (5)
#define TPCC_SUCCESS (0)
#define TRPC_ERROR (1)
#define INVALID_NEWO (100)
typedef struct  {
idl_long_int sec;
idl_long_int usec;
} time_type;
typedef struct  {
idl_short_int returncode;
idl_short_int stats;
time_type srv_start;
time_type srv_end;
time_type clnt_start;
time_type clnt_end;
} data_header;
typedef struct  {
idl_long_int first_wh;
```

```
idl_long_int last_wh;
idl_long_int server_id;
} dbInfo_data_t;

#ifdef __cplusplus
    }

#endif
#endif
```

tpcc_com_all.rgs

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

tpcc_com_all_i.c

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

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

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


 /* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jan 24 20:00:20 2000
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
        DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

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

#ifdef __cplusplus
extern "C"{
#endif


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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0
xBF,0xE0,0x8B);


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


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


MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x
4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_Payment,0xCD02F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,
0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F
,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID
```

```
#ifdef __cplusplus
}
#endif



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


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

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

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


 /* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jan 24 20:00:20 2000
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext,
robust
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

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

#ifdef __cplusplus
extern "C"{
#endif


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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0
xBF,0xE0,0x8B);


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


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


MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x
4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_Payment,0xCD02F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,
0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F
,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

tpcc_com_no.rgs

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

```
	}
}
```

tpcc_com_os.rgs

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

tpcc_com_pay.rgs

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

tpcc_com_ps.def

```
LIBRARY	"tpcc_com_ps"

DESCRIPTION	'Proxy/Stub DLL'

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

tpcc_com_ps.dsp

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

# TARGTYPE "Win32 (x86) Application" 0x0101

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

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

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
"_WINDOWS" /YX /FD /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
_WIN32_WINNT=0x0400 /D "REGISTER_PROXY_DLL" /FD /c
# SUBTRACT CPP /YX
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
```

```
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib
odbccp32.lib /nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpcrt4.lib oleaut32.lib uuid.lib
/nologo /entry:"DllMain" /subsystem:windows /dll /pdb:none /machine:I386
/def:".\src\tpcc_com_ps.def"
# Begin Custom Build - Copying tpcc_com_ps.h
InputPath=.\bin\tpcc_com_ps.dll
SOURCE="$(InputPath)"

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

# End Custom Build

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

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

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

# End Custom Build

!ENDIF

# Begin Target

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

# PROP Default_Filter ""
# Begin Source File

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

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

SOURCE=.\src\tpcc_com_ps.idl

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

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

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

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

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

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

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

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

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

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

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

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

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

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

!ENDIF

# End Source File
# Begin Source File

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

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

tpcc_com_ps.h

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

/* this ALWAYS GENERATED file contains the definitions for the interfaces */


 /* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jan 24 20:00:07 2000
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
        DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


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

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

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

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

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

/* Forward Declarations */

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


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

#ifdef __cplusplus
extern "C"{
#endif

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

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

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


EXTERN_C const IID IID_ITPCC;

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

    MIDL_INTERFACE("FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
    ITPCC : public IUnknown
    {
    public:
        virtual HRESULT __stdcall NewOrder(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR
*txn) = 0;

        virtual HRESULT __stdcall Payment(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR
*txn) = 0;

        virtual HRESULT __stdcall Delivery(
            /* [in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn)
= 0;

        virtual HRESULT __stdcall StockLevel(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR
*txn) = 0;

        virtual HRESULT __stdcall OrderStatus(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR
*txn) = 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,
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR
*txn);

        HRESULT ( __stdcall __RPC_FAR *Payment )(
            ITPCC __RPC_FAR * This,
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR
*txn);

        HRESULT ( __stdcall __RPC_FAR *Delivery )(
            ITPCC __RPC_FAR * This,
            /* [in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][in] */ unsigned char __RPC_FAR *__RPC_FAR
*txn);

        HRESULT ( __stdcall __RPC_FAR *StockLevel )(
```

```
        ITPCC __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR *iSize,
        /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR
*txn);

      HRESULT ( __stdcall __RPC_FAR *OrderStatus )(
        ITPCC __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR *iSize,
        /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR
*txn);

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

      END_INTERFACE
   } ITPCCVtbl;

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

#ifdef COBJMACROS


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

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

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


#define ITPCC_NewOrder(This,iSize,txn)    \
   (This)->lpVtbl -> NewOrder(This,iSize,txn)

#define ITPCC_Payment(This,iSize,txn)      \
   (This)->lpVtbl -> Payment(This,iSize,txn)

#define ITPCC_Delivery(This,iSize,txn)      \
   (This)->lpVtbl -> Delivery(This,iSize,txn)

#define ITPCC_StockLevel(This,iSize,txn)  \
   (This)->lpVtbl -> StockLevel(This,iSize,txn)

#define ITPCC_OrderStatus(This,iSize,txn) \
   (This)->lpVtbl -> OrderStatus(This,iSize,txn)

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

#endif /* COBJMACROS */


#endif      /* C style interface */

HRESULT __stdcall ITPCC_NewOrder_Proxy(
   ITPCC __RPC_FAR * This,
   /* [out][in] */ int __RPC_FAR *iSize,
   /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR
*txn);


void __RPC_STUB ITPCC_NewOrder_Stub(
   IRpcStubBuffer *This,
   IRpcChannelBuffer *_pRpcChannelBuffer,
   PRPC_MESSAGE _pRpcMessage,
   DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_Payment_Proxy(
   ITPCC __RPC_FAR * This,
   /* [out][in] */ int __RPC_FAR *iSize,
   /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR
*txn);

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] */ int __RPC_FAR *iSize,
   /* [size_is][size_is][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

void __RPC_STUB ITPCC_Delivery_Stub(
   IRpcStubBuffer *This,
   IRpcChannelBuffer *_pRpcChannelBuffer,
   PRPC_MESSAGE _pRpcMessage,
   DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_StockLevel_Proxy(
   ITPCC __RPC_FAR * This,
   /* [out][in] */ int __RPC_FAR *iSize,
   /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR
*txn);

void __RPC_STUB ITPCC_StockLevel_Stub(
   IRpcStubBuffer *This,
   IRpcChannelBuffer *_pRpcChannelBuffer,
   PRPC_MESSAGE _pRpcMessage,
   DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_OrderStatus_Proxy(
   ITPCC __RPC_FAR * This,
   /* [out][in] */ int __RPC_FAR *iSize,
   /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR
*txn);

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 */

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

tpcc_com_ps.idl

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

// Forward declare all types defined
//interface ITPCC;
import "oaidl.idl";
import "ocidl.idl";

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

    HRESULT _stdcall NewOrder
                                                        (
                                                        [in, out] int*
iSize,
                                                        [in, out,
size_is( , *iSize)] char** txn
                                                        );

    HRESULT _stdcall Payment
                                                        (
                                                        [in, out] int*
iSize,
                                                        [in, out,
size_is( , *iSize)] char** txn
                                                        );

    HRESULT _stdcall Delivery
                                                        (
                                                        [in] int*
iSize,
                                                        [in, size_is( ,
*iSize)] char** txn
                                                        );

    HRESULT _stdcall StockLevel
                                                        (
                                                        [in, out] int*
iSize,
                                                        [in, out,
size_is( , *iSize)] char** txn
                                                        );

    HRESULT _stdcall OrderStatus
                                                        (
                                                        [in, out] int*
iSize,
                                                        [in, out,
size_is( , *iSize)] char** txn
                                                        );

    HRESULT _stdcall CallSetComplete
                                                        (
                                                        );
    };      // interface ITPCC
```

tpcc_com_ps_i.c

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

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

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


 /* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jan 24 20:00:07 2000
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
        DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

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

#ifdef __cplusplus
extern "C"{
#endif


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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

```
#endif !_MIDL_USE_GUIDDEF_

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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

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

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

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

 /* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jan 24 20:00:07 2000
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext,
robust
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

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

#ifdef __cplusplus
extern "C"{
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

tpcc_com_ps_p.c

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

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


 /* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jan 24 20:00:07 2000
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

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


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


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


#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE   33
#define PROC_FORMAT_STRING_SIZE   193
#define TRANSMIT_AS_TABLE_SIZE    0
#define WIRE_MARSHAL_TABLE_SIZE   0

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;


/* Object interface: IUnknown, ver. 0.0,

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


/* Object interface: ITPCC, ver. 0.0,

GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0x
E0,0x8B}} */


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
};


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

#pragma data_seg(".rdata")

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

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this stub because it uses these
features:
#error   -Oif or -Oicf.
#error However, your C/C++ compilation flags indicate you intend to run this app
on earlier systems.
#error This app will die there with the RPC_X_WRONG_STUB_VERSION
error.
#endif


static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
    {
        0,
        {

            /* Procedure NewOrder */

                                    0x33,               /*
FC_AUTO_HANDLE */
                                    0x6c,               /* Old Flags:  object,
Oi2 */
/*  2 */    NdrFcLong( 0x0 ),   /* 0 */
/*  6 */    NdrFcShort( 0x3 ),  /* 3 */
#ifndef _ALPHA_
/*  8 */    NdrFcShort( 0x10 ),   /* x86, MIPS, PPC Stack size/offset = 16 */
```

```
#else
                                NdrFcShort( 0x20 ),   /* Alpha Stack
size/offset = 32 */
#endif
/* 10 */    NdrFcShort( 0x8 ),    /* 8 */
/* 12 */    NdrFcShort( 0x10 ),   /* 16 */
/* 14 */    0x7,                  /* Oi2 Flags:  srv must size, clt must size, has
return, */
                                0x3,                  /* 3 */

            /* Parameter iSize */

/* 16 */    NdrFcShort( 0x158 ), /* Flags:  in, out, base type, simple ref, */
#ifndef _ALPHA_
/* 18 */    NdrFcShort( 0x4 ),    /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                NdrFcShort( 0x8 ),    /* Alpha Stack
size/offset = 8 */
#endif
/* 20 */    0x8,                  /* FC_LONG */
                                0x0,                  /* 0 */

            /* Parameter txn */

/* 22 */    NdrFcShort( 0x201b ),/* Flags:  must size, must free, in, out, srv alloc
size=8 */
#ifndef _ALPHA_
/* 24 */    NdrFcShort( 0x8 ),    /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                NdrFcShort( 0x10 ),   /* Alpha Stack
size/offset = 16 */
#endif
/* 26 */    NdrFcShort( 0x6 ),    /* Type Offset=6 */

            /* Return value */

/* 28 */    NdrFcShort( 0x70 ),   /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 30 */    NdrFcShort( 0xc ),    /* x86, MIPS, PPC Stack size/offset = 12 */
#else
                                NdrFcShort( 0x18 ),   /* Alpha Stack
size/offset = 24 */
#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_
/* 42 */    NdrFcShort( 0x10 ),   /* x86, MIPS, PPC Stack size/offset = 16 */
#else
                                NdrFcShort( 0x20 ),   /* Alpha Stack
size/offset = 32 */
#endif
/* 44 */    NdrFcShort( 0x8 ),    /* 8 */
/* 46 */    NdrFcShort( 0x10 ),   /* 16 */
/* 48 */    0x7,                  /* Oi2 Flags:  srv must size, clt must size, has
return, */
                                0x3,                  /* 3 */

            /* Parameter iSize */

/* 50 */    NdrFcShort( 0x158 ), /* Flags:  in, out, base type, simple ref, */
#ifndef _ALPHA_
/* 52 */    NdrFcShort( 0x4 ),    /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                NdrFcShort( 0x8 ),    /* Alpha Stack
size/offset = 8 */
#endif
/* 54 */    0x8,                  /* FC_LONG */
                                0x0,                  /* 0 */

            /* Parameter txn */

/* 56 */    NdrFcShort( 0x201b ),/* Flags:  must size, must free, in, out, srv alloc
size=8 */
#ifndef _ALPHA_
/* 58 */    NdrFcShort( 0x8 ),    /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                NdrFcShort( 0x10 ),   /* Alpha Stack
size/offset = 16 */
#endif
/* 60 */    NdrFcShort( 0x6 ),    /* Type Offset=6 */

            /* Return value */

/* 62 */    NdrFcShort( 0x70 ),   /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 64 */    NdrFcShort( 0xc ),    /* x86, MIPS, PPC Stack size/offset = 12 */
#else
                                NdrFcShort( 0x18 ),   /* Alpha Stack
size/offset = 24 */
#endif
/* 66 */    0x8,                  /* FC_LONG */
                                0x0,                  /* 0 */

            /* Procedure Delivery */

/* 68 */    0x33,                 /* FC_AUTO_HANDLE */
                                0x6c,                 /* Old Flags:  object,
Oi2 */
/* 70 */    NdrFcLong( 0x0 ),     /* 0 */
/* 74 */    NdrFcShort( 0x5 ),    /* 5 */
#ifndef _ALPHA_
/* 76 */    NdrFcShort( 0x10 ),   /* x86, MIPS, PPC Stack size/offset = 16 */
#else
                                NdrFcShort( 0x20 ),   /* Alpha Stack
size/offset = 32 */
#endif
/* 78 */    NdrFcShort( 0x8 ),    /* 8 */
/* 80 */    NdrFcShort( 0x8 ),    /* 8 */
/* 82 */    0x6,                  /* Oi2 Flags:  clt must size, has return, */
                                0x3,                  /* 3 */

            /* Parameter iSize */

/* 84 */    NdrFcShort( 0x148 ), /* Flags:  in, base type, simple ref, */
#ifndef _ALPHA_
/* 86 */    NdrFcShort( 0x4 ),    /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                NdrFcShort( 0x8 ),    /* Alpha Stack
size/offset = 8 */
#endif
/* 88 */    0x8,                  /* FC_LONG */
                                0x0,                  /* 0 */

            /* Parameter txn */

/* 90 */    NdrFcShort( 0x200b ),/* Flags:  must size, must free, in, srv alloc
size=8 */
#ifndef _ALPHA_
/* 92 */    NdrFcShort( 0x8 ),    /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                NdrFcShort( 0x10 ),   /* Alpha Stack
size/offset = 16 */
#endif
/* 94 */    NdrFcShort( 0x18 ),   /* Type Offset=24 */
```

```
            /* Return value */

/* 96 */    NdrFcShort( 0x70 ),   /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 98 */    NdrFcShort( 0xc ),    /* x86, MIPS, PPC Stack size/offset = 12 */
#else
                                  NdrFcShort( 0x18 ),   /* Alpha Stack
size/offset = 24 */
#endif
/* 100 */   0x8,                  /* FC_LONG */
                                  0x0,                  /* 0 */

            /* Procedure StockLevel */

/* 102 */   0x33,                 /* FC_AUTO_HANDLE */
                                  0x6c,                 /* Old Flags:  object,
Oi2 */
/* 104 */   NdrFcLong( 0x0 ),     /* 0 */
/* 108 */   NdrFcShort( 0x6 ),    /* 6 */
#ifndef _ALPHA_
/* 110 */   NdrFcShort( 0x10 ),   /* x86, MIPS, PPC Stack size/offset = 16 */
#else
                                  NdrFcShort( 0x20 ),   /* Alpha Stack
size/offset = 32 */
#endif
/* 112 */   NdrFcShort( 0x8 ),    /* 8 */
/* 114 */   NdrFcShort( 0x10 ),   /* 16 */
/* 116 */   0x7,                  /* Oi2 Flags:  srv must size, clt must size, has
return, */
                                  0x3,                  /* 3 */

            /* Parameter iSize */

/* 118 */   NdrFcShort( 0x158 ), /* Flags:  in, out, base type, simple ref, */
#ifndef _ALPHA_
/* 120 */   NdrFcShort( 0x4 ),    /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                  NdrFcShort( 0x8 ),    /* Alpha Stack
size/offset = 8 */
#endif
/* 122 */   0x8,                  /* FC_LONG */
                                  0x0,                  /* 0 */

            /* Parameter txn */

/* 124 */   NdrFcShort( 0x201b ),/* Flags:  must size, must free, in, out, srv alloc
size=8 */
#ifndef _ALPHA_
/* 126 */   NdrFcShort( 0x8 ),    /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                  NdrFcShort( 0x10 ),   /* Alpha Stack
size/offset = 16 */
#endif
/* 128 */   NdrFcShort( 0x6 ),    /* Type Offset=6 */

            /* Return value */

/* 130 */   NdrFcShort( 0x70 ),   /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 132 */   NdrFcShort( 0xc ),    /* x86, MIPS, PPC Stack size/offset = 12 */
#else
                                  NdrFcShort( 0x18 ),   /* Alpha Stack
size/offset = 24 */
#endif
/* 134 */   0x8,                  /* FC_LONG */
                                  0x0,                  /* 0 */

            /* Procedure OrderStatus */

/* 136 */   0x33,                 /* FC_AUTO_HANDLE */
                                  0x6c,                 /* Old Flags:  object,
Oi2 */
/* 138 */   NdrFcLong( 0x0 ),     /* 0 */
/* 142 */   NdrFcShort( 0x7 ),    /* 7 */
#ifndef _ALPHA_
/* 144 */   NdrFcShort( 0x10 ),   /* x86, MIPS, PPC Stack size/offset = 16 */
#else
                                  NdrFcShort( 0x20 ),   /* Alpha Stack
size/offset = 32 */
#endif
/* 146 */   NdrFcShort( 0x8 ),    /* 8 */
/* 148 */   NdrFcShort( 0x10 ),   /* 16 */
/* 150 */   0x7,                  /* Oi2 Flags:  srv must size, clt must size, has
return, */
                                  0x3,                  /* 3 */

            /* Parameter iSize */

/* 152 */   NdrFcShort( 0x158 ), /* Flags:  in, out, base type, simple ref, */
#ifndef _ALPHA_
/* 154 */   NdrFcShort( 0x4 ),    /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                  NdrFcShort( 0x8 ),    /* Alpha Stack
size/offset = 8 */
#endif
/* 156 */   0x8,                  /* FC_LONG */
                                  0x0,                  /* 0 */

            /* Parameter txn */

/* 158 */   NdrFcShort( 0x201b ),/* Flags:  must size, must free, in, out, srv alloc
size=8 */
#ifndef _ALPHA_
/* 160 */   NdrFcShort( 0x8 ),    /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                  NdrFcShort( 0x10 ),   /* Alpha Stack
size/offset = 16 */
#endif
/* 162 */   NdrFcShort( 0x6 ),    /* Type Offset=6 */

            /* Return value */

/* 164 */   NdrFcShort( 0x70 ),   /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 166 */   NdrFcShort( 0xc ),    /* x86, MIPS, PPC Stack size/offset = 12 */
#else
                                  NdrFcShort( 0x18 ),   /* Alpha Stack
size/offset = 24 */
#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 */
                                  0x11, 0x8, /* FC_RP [simple_pointer] */
/*  4 */    0x8,                  /* FC_LONG */
                                  0x5c,                 /* FC_PAD */
/*  6 */
                                  0x11, 0x14,           /* FC_RP
[alloced_on_stack] [pointer_deref] */
/*  8 */    NdrFcShort( 0x2 ),    /* Offset= 2 (10) */
/* 10 */
                                  0x13, 0x0, /* FC_OP */
/* 12 */    NdrFcShort( 0x2 ),    /* Offset= 2 (14) */
/* 14 */
                                  0x1b,                 /* FC_CARRAY */
                                  0x0,                  /* 0 */
/* 16 */    NdrFcShort( 0x1 ),    /* 1 */
/* 18 */    0x28,                 /* Corr desc:  parameter, FC_LONG */
                                  0x54,                 /* FC_DEREFERENCE
*/
#ifndef _ALPHA_
/* 20 */    NdrFcShort( 0x4 ),    /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                  NdrFcShort( 0x8 ),    /* Alpha Stack
size/offset = 8 */
#endif
/* 22 */    0x2,                  /* FC_CHAR */
                                  0x5b,                 /* FC_END */
/* 24 */
                                  0x11, 0x14,           /* FC_RP
[alloced_on_stack] [pointer_deref] */
/* 26 */    NdrFcShort( 0x2 ),    /* Offset= 2 (28) */
/* 28 */
                                  0x12, 0x0, /* FC_UP */
/* 30 */    NdrFcShort( 0xfffffff0 ),           /* Offset= -16 (14) */

                                  0x0
        }
    };

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

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

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


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

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{

    if(!_tpcc_com_ps_CHECK_IID(0))
        {
        *pIndex = 0;
        return 1;
        }

    return 0;
}

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


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


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

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


 /* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jan 24 20:00:07 2000
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext,
robust
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

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


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


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

```
#endif // __RPCPROXY_H_VERSION__


#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE   35
#define PROC_FORMAT_STRING_SIZE   253
#define TRANSMIT_AS_TABLE_SIZE    0
#define WIRE_MARSHAL_TABLE_SIZE   0

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;


/* Object interface: IUnknown, ver. 0.0,

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


/* Object interface: ITPCC, ver. 0.0,

GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0x
E0,0x8B}} */


extern const MIDL_STUB_DESC Object_StubDesc;


extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[] =
    {
    0,
    44,
    88,
    132,
    176,
    220
    };

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

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

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

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


static const MIDL_STUB_DESC Object_StubDesc =
    {
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x50002, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.3.280 */
    0,
    0,
    0,  /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0,  /* Reserved3 */
    0,  /* Reserved4 */
    0   /* Reserved5 */
    };

#pragma data_seg(".rdata")

#if !defined(__RPC_WIN64__)
#error  Invalid build platform for this stub.
#endif

static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
    {
        0,
        {

            /* Procedure NewOrder */

                                    0x33,               /*
FC_AUTO_HANDLE */
```

```
                                0x6c,                   /* Old Flags:  object,
Oi2 */
/*  2 */    NdrFcLong( 0x0 ),    /* 0 */
/*  6 */    NdrFcShort( 0x3 ),   /* 3 */
/*  8 */    NdrFcShort( 0x20 ),  /* ia64, axp64 Stack size/offset = 32 */
/* 10 */    NdrFcShort( 0x8 ),   /* 8 */
/* 12 */    NdrFcShort( 0x10 ),  /* 16 */
/* 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( 0x1 ),   /* 1 */
/* 20 */    NdrFcShort( 0x1 ),   /* 1 */
/* 22 */    NdrFcShort( 0x0 ),   /* 0 */
/* 24 */    NdrFcShort( 0x0 ),   /* 0 */

            /* Parameter iSize */

/* 26 */    NdrFcShort( 0x158 ), /* Flags:  in, out, base type, simple ref, */
/* 28 */    NdrFcShort( 0x8 ),   /* ia64, axp64 Stack size/offset = 8 */
/* 30 */    0x8,                 /* FC_LONG */
                                0x0,                    /* 0 */

            /* Parameter txn */

/* 32 */    NdrFcShort( 0x201b ),/* Flags:  must size, must free, in, out, srv alloc
size=8 */
/* 34 */    NdrFcShort( 0x10 ),  /* ia64, axp64 Stack size/offset = 16 */
/* 36 */    NdrFcShort( 0x6 ),   /* Type Offset=6 */

            /* Return value */

/* 38 */    NdrFcShort( 0x70 ),  /* Flags:  out, return, base type, */
/* 40 */    NdrFcShort( 0x18 ),  /* ia64, axp64 Stack size/offset = 24 */
/* 42 */    0x8,                 /* FC_LONG */
                                0x0,                    /* 0 */

            /* Procedure Payment */

/* 44 */    0x33,                /* FC_AUTO_HANDLE */
                                0x6c,                   /* Old Flags:  object,
Oi2 */
/* 46 */    NdrFcLong( 0x0 ),    /* 0 */
/* 50 */    NdrFcShort( 0x4 ),   /* 4 */
/* 52 */    NdrFcShort( 0x20 ),  /* ia64, axp64 Stack size/offset = 32 */
/* 54 */    NdrFcShort( 0x8 ),   /* 8 */
/* 56 */    NdrFcShort( 0x10 ),  /* 16 */
/* 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( 0x1 ),   /* 1 */
/* 64 */    NdrFcShort( 0x1 ),   /* 1 */
/* 66 */    NdrFcShort( 0x0 ),   /* 0 */
/* 68 */    NdrFcShort( 0x0 ),   /* 0 */

            /* Parameter iSize */

/* 70 */    NdrFcShort( 0x158 ), /* Flags:  in, out, base type, simple ref, */
/* 72 */    NdrFcShort( 0x8 ),   /* ia64, axp64 Stack size/offset = 8 */
/* 74 */    0x8,                 /* FC_LONG */
                                0x0,                    /* 0 */

            /* Parameter txn */

/* 76 */    NdrFcShort( 0x201b ),/* Flags:  must size, must free, in, out, srv alloc
size=8 */
/* 78 */    NdrFcShort( 0x10 ),  /* ia64, axp64 Stack size/offset = 16 */
/* 80 */    NdrFcShort( 0x6 ),   /* Type Offset=6 */

            /* Return value */

/* 82 */    NdrFcShort( 0x70 ),  /* Flags:  out, return, base type, */
/* 84 */    NdrFcShort( 0x18 ),  /* ia64, axp64 Stack size/offset = 24 */
/* 86 */    0x8,                 /* FC_LONG */
                                0x0,                    /* 0 */

            /* Procedure Delivery */

/* 88 */    0x33,                /* FC_AUTO_HANDLE */
                                0x6c,                   /* Old Flags:  object,
Oi2 */
/* 90 */    NdrFcLong( 0x0 ),    /* 0 */
/* 94 */    NdrFcShort( 0x5 ),   /* 5 */
/* 96 */    NdrFcShort( 0x20 ),  /* ia64, axp64 Stack size/offset = 32 */
/* 98 */    NdrFcShort( 0x8 ),   /* 8 */
/* 100 */   NdrFcShort( 0x8 ),   /* 8 */
/* 102 */   0x46,                /* Oi2 Flags:  clt must size, has return, has ext,
*/
                                0x3,                    /* 3 */
/* 104 */   0xa,                 /* 10 */
                                0x5,                    /* Ext Flags:  new corr
desc, srv corr check, */
/* 106 */   NdrFcShort( 0x0 ),   /* 0 */
/* 108 */   NdrFcShort( 0x1 ),   /* 1 */
/* 110 */   NdrFcShort( 0x0 ),   /* 0 */
/* 112 */   NdrFcShort( 0x0 ),   /* 0 */

            /* Parameter iSize */

/* 114 */   NdrFcShort( 0x148 ), /* Flags:  in, base type, simple ref, */
/* 116 */   NdrFcShort( 0x8 ),   /* ia64, axp64 Stack size/offset = 8 */
/* 118 */   0x8,                 /* FC_LONG */
                                0x0,                    /* 0 */

            /* Parameter txn */

/* 120 */   NdrFcShort( 0x200b ),/* Flags:  must size, must free, in, srv alloc
size=8 */
/* 122 */   NdrFcShort( 0x10 ),  /* ia64, axp64 Stack size/offset = 16 */
/* 124 */   NdrFcShort( 0x1a ),  /* Type Offset=26 */

            /* Return value */

/* 126 */   NdrFcShort( 0x70 ),  /* Flags:  out, return, base type, */
/* 128 */   NdrFcShort( 0x18 ),  /* ia64, axp64 Stack size/offset = 24 */
/* 130 */   0x8,                 /* FC_LONG */
                                0x0,                    /* 0 */

            /* Procedure StockLevel */

/* 132 */   0x33,                /* FC_AUTO_HANDLE */
                                0x6c,                   /* Old Flags:  object,
Oi2 */
/* 134 */   NdrFcLong( 0x0 ),    /* 0 */
/* 138 */   NdrFcShort( 0x6 ),   /* 6 */
/* 140 */   NdrFcShort( 0x20 ),  /* ia64, axp64 Stack size/offset = 32 */
/* 142 */   NdrFcShort( 0x8 ),   /* 8 */
/* 144 */   NdrFcShort( 0x10 ),  /* 16 */
/* 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( 0x1 ),   /* 1 */
/* 152 */   NdrFcShort( 0x1 ),   /* 1 */
/* 154 */   NdrFcShort( 0x0 ),   /* 0 */
```

```
/* 156 */   NdrFcShort( 0x0 ),      /* 0 */

            /* Parameter iSize */

/* 158 */   NdrFcShort( 0x158 ), /* Flags:  in, out, base type, simple ref, */
/* 160 */   NdrFcShort( 0x8 ),      /* ia64, axp64 Stack size/offset = 8 */
/* 162 */   0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Parameter txn */

/* 164 */   NdrFcShort( 0x201b ),/* Flags:  must size, must free, in, out, srv alloc
size=8 */
/* 166 */   NdrFcShort( 0x10 ),     /* ia64, axp64 Stack size/offset = 16 */
/* 168 */   NdrFcShort( 0x6 ),      /* Type Offset=6 */

            /* Return value */

/* 170 */   NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
/* 172 */   NdrFcShort( 0x18 ),     /* ia64, axp64 Stack size/offset = 24 */
/* 174 */   0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Procedure OrderStatus */

/* 176 */   0x33,                   /* FC_AUTO_HANDLE */
                                    0x6c,                   /* Old Flags:  object,
Oi2 */
/* 178 */   NdrFcLong( 0x0 ),       /* 0 */
/* 182 */   NdrFcShort( 0x7 ),      /* 7 */
/* 184 */   NdrFcShort( 0x20 ),     /* ia64, axp64 Stack size/offset = 32 */
/* 186 */   NdrFcShort( 0x8 ),      /* 8 */
/* 188 */   NdrFcShort( 0x10 ),     /* 16 */
/* 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( 0x1 ),      /* 1 */
/* 196 */   NdrFcShort( 0x1 ),      /* 1 */
/* 198 */   NdrFcShort( 0x0 ),      /* 0 */
/* 200 */   NdrFcShort( 0x0 ),      /* 0 */

            /* Parameter iSize */

/* 202 */   NdrFcShort( 0x158 ), /* Flags:  in, out, base type, simple ref, */
/* 204 */   NdrFcShort( 0x8 ),      /* ia64, axp64 Stack size/offset = 8 */
/* 206 */   0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Parameter txn */

/* 208 */   NdrFcShort( 0x201b ),/* Flags:  must size, must free, in, out, srv alloc
size=8 */
/* 210 */   NdrFcShort( 0x10 ),     /* ia64, axp64 Stack size/offset = 16 */
/* 212 */   NdrFcShort( 0x6 ),      /* Type Offset=6 */

            /* Return value */

/* 214 */   NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
/* 216 */   NdrFcShort( 0x18 ),     /* ia64, axp64 Stack size/offset = 24 */
/* 218 */   0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Procedure CallSetComplete */

/* 220 */   0x33,                   /* FC_AUTO_HANDLE */
                                    0x6c,                   /* Old Flags:  object,
Oi2 */
/* 222 */   NdrFcLong( 0x0 ),       /* 0 */
/* 226 */   NdrFcShort( 0x8 ),      /* 8 */
/* 228 */   NdrFcShort( 0x10 ),     /* ia64, axp64 Stack size/offset = 16 */
/* 230 */   NdrFcShort( 0x0 ),      /* 0 */
/* 232 */   NdrFcShort( 0x8 ),      /* 8 */
/* 234 */   0x44,                   /* Oi2 Flags:  has return, has ext, */
                                    0x1,                    /* 1 */
/* 236 */   0xa,                    /* 10 */
                                    0x1,                    /* Ext Flags:  new corr
desc, */
/* 238 */   NdrFcShort( 0x0 ),      /* 0 */
/* 240 */   NdrFcShort( 0x0 ),      /* 0 */
/* 242 */   NdrFcShort( 0x0 ),      /* 0 */
/* 244 */   NdrFcShort( 0x0 ),      /* 0 */

            /* Return value */

/* 246 */   NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
/* 248 */   NdrFcShort( 0x8 ),      /* ia64, axp64 Stack size/offset = 8 */
/* 250 */   0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */


                                    0x0
        }
    };

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
    {
        0,
        {
                                    NdrFcShort( 0x0 ),      /* 0 */
/*  2 */
                                    0x11, 0x8, /* FC_RP [simple_pointer] */
/*  4 */    0x8,                    /* FC_LONG */
                                    0x5c,                   /* FC_PAD */
/*  6 */
                                    0x11, 0x14,             /* FC_RP
[alloced_on_stack] [pointer_deref] */
/*  8 */    NdrFcShort( 0x2 ),      /* Offset= 2 (10) */
/* 10 */
                                    0x13, 0x0, /* FC_OP */
/* 12 */    NdrFcShort( 0x2 ),      /* Offset= 2 (14) */
/* 14 */
                                    0x1b,                   /* FC_CARRAY */
                                    0x0,                    /* 0 */
/* 16 */    NdrFcShort( 0x1 ),      /* 1 */
/* 18 */    0x28,                   /* Corr desc:  parameter, FC_LONG */
                                    0x54,                   /* FC_DEREFERENCE
*/
/* 20 */    NdrFcShort( 0x8 ),      /* ia64, axp64 Stack size/offset = 8 */
/* 22 */    NdrFcShort( 0x1 ),      /* Corr flags:  early, */
/* 24 */    0x2,                    /* FC_CHAR */
                                    0x5b,                   /* FC_END */
/* 26 */
                                    0x11, 0x14,             /* FC_RP
[alloced_on_stack] [pointer_deref] */
/* 28 */    NdrFcShort( 0x2 ),      /* Offset= 2 (30) */
/* 30 */
                                    0x12, 0x0, /* FC_UP */
/* 32 */    NdrFcShort( 0xfffffffe ),       /* Offset= -18 (14) */

                                    0x0
        }
    };

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

const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[] =
```

```
{
   ( CInterfaceStubVtbl *) &_ITPCCStubVtbl,
   0
};

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


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

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{

   if(!_tpcc_com_ps_CHECK_IID(0))
      {
      *pIndex = 0;
      return 1;
      }

   return 0;
}

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


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

tpcc_com_sl.rgs

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

tpcc_dblib.cpp

```
/*          FILE:                   TPCC_DBLIB.CPP
 *                                              Microsoft TPC-C Kit Ver. 4.20.000
 *                                              Copyright Microsoft, 1999
 *          All Rights Reserved
 *
 *                                              Version 4.10.000 audited by
Richard Gimarc, Performance Metrics, 3/17/99
 *
 *          PURPOSE:            Implements dblib calls for TPC-C txns.
 *          Contact:    Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                      4.20.000 - updated rev number to match kit
 *                      4.10.001 - not deleting error class in catch handler on
deadlock retry;
 *                                                  not a functional bug, but a
memory leak
 *                                                  - had to tweak some declarations to
compile with latest SDK; no functional change
 */

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

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

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

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

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

#define     DEFCLPACKSIZE                       4096

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

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

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

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

```
                        break;

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

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


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

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

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

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

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

int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int severity,

                                        LPCSTR msgtext, LPCSTR
srvname, LPCSTR procname, DBUSMALLINT line)
{
        CTPCC_DBLIB                             *pConn;

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

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

        return 0;
}

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

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

        return;
}

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
 *
 */

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

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

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

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

```
                return errorMsgs[i].szMsg;
}

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


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

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

        m_MaxRetries = 10;          // how many retries on deadlock

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

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

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

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


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

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

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

        m_dbproc = dbopen(login, szServer);

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

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

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

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

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

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

        DiscardNextResults(2);

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

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

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

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

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


        DiscardNextRows(0);
        DiscardNextResults(0);
}


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

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

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

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

}

void CTPCC_DBLIB::SetSqlError( int /*DBINT*/ msgno, int msgstate, int
severity, LPCSTR msgtext )
{
	if (m_SqlErr == NULL)
		m_SqlErr = new CSQLERR();

	m_SqlErr->m_msgno = msgno;
	m_SqlErr->m_msgstate = msgstate;
	m_SqlErr->m_severity = severity;

	delete [] m_SqlErr->m_msgtext;
	if (msgtext != NULL)
	{
		m_SqlErr->m_msgtext = new char[ strlen(msgtext)+1 ];
		strcpy( m_SqlErr->m_msgtext, msgtext );
	}
}

void CTPCC_DBLIB::ThrowError( CDBLIBERR::ACTION eAction )
{
	// discard anything still in return buffer
	DiscardNextRows(-1);
	DiscardNextResults(-1);

	// check for SQL Server error first;  if yes, throw it and ignore any
DBLib error.
	if (m_SqlErr != NULL)
	{
		CSQLERR		*pSqlErr;
		pSqlErr = m_SqlErr;
		m_SqlErr = NULL;    // clear our pointer to instance; catch
handler will delete
		throw pSqlErr;
	}

	CDBLIBERR		*pDbLibErr;
	if (m_DbLibErr == NULL)
		// this case isn't expected to happen, since it means that an
error was returned
		// but the error handlers were not called.
		pDbLibErr = new CDBLIBERR(eAction);
	else
	{
		pDbLibErr = m_DbLibErr;
		pDbLibErr->m_eAction = eAction;
		m_DbLibErr = NULL;				// clear our
pointer to instance; catch handler will delete
	}

	throw pDbLibErr;
}

// Read and discard rows until no more.  Throw an exception if number of rows
read doesn't
// match number of rows expected.  The row count will be ignored if the expected
count value
// passed in is negative.  A typical use of this routine is to verify that there are no
more
// rows to be read.
void CTPCC_DBLIB::DiscardNextRows(int iExpectedCount)
{
	int			iRowsRead = 0;
	RETCODE		rc;

	while (TRUE)
	{
		rc = dbnextrow(m_dbproc);
		if (rc == NO_MORE_ROWS)
			break;
		if (rc == FAIL)
		{
			if (iExpectedCount >= 0)

ThrowError(CDBLIBERR::eDbNextRow);
			else
				break;
		}
		iRowsRead++;
	}

	if ((iExpectedCount >= 0) &&
		(iExpectedCount != iRowsRead))
		ThrowError(CDBLIBERR::eWrongRowCount);
}

// Read and discard results until no more.  Throw an exception if number of result
sets read doesn't
// match number expected.  The result set count will be ignored if the expected
count value
// passed in is negative.  A typical use of this routine is to verify that there are no
more
// result sets to be read.
void CTPCC_DBLIB::DiscardNextResults(int iExpectedCount)
{
	int			iResultsRead = 0;
	RETCODE		rc;

	while (TRUE)
	{
		rc = dbresults(m_dbproc);
		if (rc == NO_MORE_RESULTS)
			break;
		if (rc == FAIL)
		{
			if (iExpectedCount >= 0)

ThrowError(CDBLIBERR::eDbResults);
			else
				break;
		}

		DiscardNextRows(-1);
		iResultsRead++;
	}

	if ((iExpectedCount >= 0) &&
		(iExpectedCount != iResultsRead))
		ThrowError(CDBLIBERR::eWrongRowCount);
}

void CTPCC_DBLIB::StockLevel()
{
	int				iTryCount = 0;
	const BYTE		*pData;

	ResetError();
```

```
	while (TRUE)
	{
		try
		{
			dbrpcinit(m_dbproc, "tpcc_stocklevel", 0);

			dbrpcparam(m_dbproc, NULL, 0, SQLINT2,
-1, -1, (BYTE *) &m_txn.StockLevel.w_id);		// @w_id smallint
			dbrpcparam(m_dbproc, NULL, 0, SQLINT1,
-1, -1, (BYTE *) &m_txn.StockLevel.d_id);		// @d_id tinyint
			dbrpcparam(m_dbproc, NULL, 0, SQLINT2,
-1, -1, (BYTE *) &m_txn.StockLevel.threshold);	// @threshhold smallint

			if (dbrpcexec(m_dbproc) == FAIL)

ThrowError(CDBLIBERR::eDbRpcExec);

			if (dbresults(m_dbproc) != SUCCEED)

ThrowError(CDBLIBERR::eDbResults);

			if (dbnextrow(m_dbproc) != REG_ROW)

ThrowError(CDBLIBERR::eDbNextRow);

			if (pData=dbdata(m_dbproc, 1))
				m_txn.StockLevel.low_stock =
*((long *) pData);

			DiscardNextRows(0);
			DiscardNextResults(0);

			m_txn.StockLevel.exec_status_code = eOK;
			return;
		}
		catch (CSQLERR *e)
		{
			if ((e->m_msgno == 1205 ||
				(e->m_msgno ==
iErrOleDbProvider &&
				strstr(e->m_msgtext,
sErrTimeoutExpired) != NULL)) &&
				(++iTryCount <= iMaxRetries))
			{
				// hit deadlock; backoff for
increasingly longer period
				delete e;
				Sleep(10 * iTryCount);
			}
			else
				throw;
		}
	}		// while (TRUE)

	//if (iTryCount)
	//		throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}


void CTPCC_DBLIB::NewOrder()
{
	int			i;
	DBINT		commit_flag;
	DBDATETIME	datetime;
	DBDATEREC	daterec;

	int			iTryCount = 0;
	const BYTE	*pData;

	ResetError();

	while (TRUE)
	{
		try
		{
			dbrpcinit(m_dbproc, "tpcc_neworder", 0);

			dbrpcparam(m_dbproc, NULL, 0, SQLINT2,
-1, -1, (BYTE *) &m_txn.NewOrder.w_id);
			dbrpcparam(m_dbproc, NULL, 0, SQLINT1,
-1, -1, (BYTE *) &m_txn.NewOrder.d_id);
			dbrpcparam(m_dbproc, NULL, 0, SQLINT4,
-1, -1, (BYTE *) &m_txn.NewOrder.c_id);
			dbrpcparam(m_dbproc, NULL, 0, SQLINT1,
-1, -1, (BYTE *) &m_txn.NewOrder.o_ol_cnt);

			// check whether any order lines are for a
remote warehouse
			m_txn.NewOrder.o_all_local = 1;
			for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++)
			{
				if
(m_txn.NewOrder.OL[i].ol_supply_w_id != m_txn.NewOrder.w_id)
				{

m_txn.NewOrder.o_all_local = 0;  // at least one remote warehouse
					break;
				}
			}
			dbrpcparam(m_dbproc, NULL, 0, SQLINT1,
-1, -1, (BYTE *) &m_txn.NewOrder.o_all_local);

			for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++)
			{
				dbrpcparam(m_dbproc, NULL, 0,
SQLINT4, -1, -1, (BYTE *) &m_txn.NewOrder.OL[i].ol_i_id);
				dbrpcparam(m_dbproc, NULL, 0,
SQLINT2, -1, -1, (BYTE *) &m_txn.NewOrder.OL[i].ol_supply_w_id);
				dbrpcparam(m_dbproc, NULL, 0,
SQLINT2, -1, -1, (BYTE *) &m_txn.NewOrder.OL[i].ol_quantity);
			}

			if (dbrpcexec(m_dbproc) == FAIL)

ThrowError(CDBLIBERR::eDbRpcExec);


			// Get order line results
			m_txn.NewOrder.total_amount = 0;

			for (i = 0; i<m_txn.NewOrder.o_ol_cnt; i++)
			{
				if (dbresults(m_dbproc) !=
SUCCEED)

ThrowError(CDBLIBERR::eDbResults);

				if (dbnumcols(m_dbproc) != 5)

ThrowError(CDBLIBERR::eWrongNumCols);

				if (dbnextrow(m_dbproc) !=
REG_ROW)

ThrowError(CDBLIBERR::eDbNextRow);

				if(pData=dbdata(m_dbproc, 1))

UtilStrCpy(m_txn.NewOrder.OL[i].ol_i_name, pData, dbdatlen(m_dbproc, 1));
				if(pData=dbdata(m_dbproc, 2))
```

```
m_txn.NewOrder.OL[i].ol_stock = (*(DBSMALLINT *) pData);
                                        if(pData=dbdata(m_dbproc, 3))
UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_generic, pData, dbdatlen(m_dbproc,
3));
                                        if(pData=dbdata(m_dbproc, 4))
                                                dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,4),
                                                SQLFLT8, (BYTE
*)&m_txn.NewOrder.OL[i].ol_i_price, 8);
                                        if(pData=dbdata(m_dbproc, 5))

                                                dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,5),
                                                SQLFLT8, (BYTE
*)&m_txn.NewOrder.OL[i].ol_amount, 8);

                                        m_txn.NewOrder.total_amount =
m_txn.NewOrder.total_amount + m_txn.NewOrder.OL[i].ol_amount;

                                        DiscardNextRows(0);
                                }


                                // get remaining values for w_tax, d_tax, o_id,
c_last, c_discount, c_credit, o_entry_d, commit_flag
                                if (dbresults(m_dbproc) != SUCCEED)

ThrowError(CDBLIBERR::eDbResults);

                                if (dbnextrow(m_dbproc) != REG_ROW)

ThrowError(CDBLIBERR::eDbNextRow);

                                if (dbnumcols(m_dbproc) != 8)

ThrowError(CDBLIBERR::eWrongNumCols);

                                if (pData=dbdata(m_dbproc, 1))


                                        dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,1), SQLFLT8, (BYTE
*)&m_txn.NewOrder.w_tax, 8);
                                if (pData=dbdata(m_dbproc, 2))

                                        dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,2), SQLFLT8, (BYTE
*)&m_txn.NewOrder.d_tax, 8);
                                if (pData=dbdata(m_dbproc, 3))
                                        m_txn.NewOrder.o_id = (*(DBINT
*) pData);
                                if (pData=dbdata(m_dbproc, 4))

UtilStrCpy(m_txn.NewOrder.c_last, pData, dbdatlen(m_dbproc, 4));
                                if (pData=dbdata(m_dbproc, 5))
                                        dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,5), SQLFLT8, (BYTE
*)&m_txn.NewOrder.c_discount, 8);
                                if (pData=dbdata(m_dbproc, 6))

UtilStrCpy(m_txn.NewOrder.c_credit, pData, dbdatlen(m_dbproc, 6));
                                if (pData=dbdata(m_dbproc, 7))
                                {
                                        datetime = *((DBDATETIME *)
pData);
                                        dbdatecrack(m_dbproc, &daterec,
&datetime);
                                        m_txn.NewOrder.o_entry_d.year
= daterec.year;
                                        m_txn.NewOrder.o_entry_d.month
= daterec.month;
                                        m_txn.NewOrder.o_entry_d.day
= daterec.day;
                                        m_txn.NewOrder.o_entry_d.hour
= daterec.hour;
                                        m_txn.NewOrder.o_entry_d.minute
= daterec.minute;
                                        m_txn.NewOrder.o_entry_d.second
= daterec.second;
                                }
                                if (pData=dbdata(m_dbproc, 8))
                                        commit_flag = (*(DBTINYINT *)
pData);

                                DiscardNextRows(0);
                                DiscardNextResults(0);


                                if (commit_flag == 1)
                                {
                                        m_txn.NewOrder.total_amount *=
((1 + m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));
                                        m_txn.NewOrder.exec_status_code
= eOK;
                                }
                                else
                                        m_txn.NewOrder.exec_status_code
= eInvalidItem;

                                return;
                        }
                        catch (CSQLERR *e)
                        {
                                if ((e->m_msgno == 1205 ||
                                        (e->m_msgno ==
iErrOleDbProvider &&
                                        strstr(e->m_msgtext,
sErrTimeoutExpired) != NULL)) &&
                                        (++iTryCount <= iMaxRetries))
                                {
                                        // hit deadlock; backoff for
increasingly longer period
                                        delete e;
                                        Sleep(10 * iTryCount);
                                }
                                else
                                        throw;
                        }
                }       // while (TRUE)

//              if (iTryCount)
//                      throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}


void CTPCC_DBLIB::Payment()
{
                DBDATETIME      datetime;
                DBDATEREC       daterec;

                int                             iTryCount = 0;
                const BYTE      *pData;

                ResetError();

                while (TRUE)
                {
```

```
		try
		{
			dbrpcinit(m_dbproc, "tpcc_payment", 0);

			dbrpcparam(m_dbproc, NULL, 0, SQLINT2,
-1, -1, (BYTE *) &m_txn.Payment.w_id);
			dbrpcparam(m_dbproc, NULL, 0, SQLINT2,
-1, -1, (BYTE *) &m_txn.Payment.c_w_id);
			dbrpcparam(m_dbproc, NULL, 0, SQLFLT8,
-1, -1, (BYTE *) &m_txn.Payment.h_amount);
			dbrpcparam(m_dbproc, NULL, 0, SQLINT1,
-1, -1, (BYTE *) &m_txn.Payment.d_id);
			dbrpcparam(m_dbproc, NULL, 0, SQLINT1,
-1, -1, (BYTE *) &m_txn.Payment.c_d_id);
			dbrpcparam(m_dbproc, NULL, 0, SQLINT4,
-1, -1, (BYTE *) &m_txn.Payment.c_id);

			// if customer id is zero, then payment is by
name
			if (m_txn.Payment.c_id == 0)
				dbrpcparam(m_dbproc, NULL, 0,
SQLCHAR, -1, strlen(m_txn.Payment.c_last), (unsigned char
*)m_txn.Payment.c_last);

			if (dbrpcexec(m_dbproc) == FAIL)

ThrowError(CDBLIBERR::eDbRpcExec);

			if (dbresults(m_dbproc) != SUCCEED)

ThrowError(CDBLIBERR::eDbResults);

			if (dbnextrow(m_dbproc) != REG_ROW)

ThrowError(CDBLIBERR::eDbNextRow);

			if (dbnumcols(m_dbproc) != 27)

ThrowError(CDBLIBERR::eWrongNumCols);

			if (pData=dbdata(m_dbproc, 1))
				m_txn.Payment.c_id = *((DBINT
*) pData);
			if (pData=dbdata(m_dbproc, 2))
				UtilStrCpy(m_txn.Payment.c_last,
pData, dbdatlen(m_dbproc, 2));
			if (pData=dbdata(m_dbproc, 3))
			{
				datetime = *((DBDATETIME *)
pData);
				dbdatecrack(m_dbproc, &daterec,
&datetime);
				m_txn.Payment.h_date.year   =
daterec.year;
				m_txn.Payment.h_date.month  =
daterec.month;
				m_txn.Payment.h_date.day    =
daterec.day;
				m_txn.Payment.h_date.hour   =
daterec.hour;
				m_txn.Payment.h_date.minute =
daterec.minute;
				m_txn.Payment.h_date.second =
daterec.second;
			}
			if (pData=dbdata(m_dbproc, 4))

UtilStrCpy(m_txn.Payment.w_street_1, pData, dbdatlen(m_dbproc, 4));
			if (pData=dbdata(m_dbproc, 5))

UtilStrCpy(m_txn.Payment.w_street_2, pData, dbdatlen(m_dbproc, 5));
			if (pData=dbdata(m_dbproc, 6))
				UtilStrCpy(m_txn.Payment.w_city,
pData, dbdatlen(m_dbproc, 6));
			if (pData=dbdata(m_dbproc, 7))
				UtilStrCpy(m_txn.Payment.w_state,
pData, dbdatlen(m_dbproc, 7));
			if (pData=dbdata(m_dbproc, 8))
				UtilStrCpy(m_txn.Payment.w_zip,
pData, dbdatlen(m_dbproc, 8));
			if (pData=dbdata(m_dbproc, 9))

UtilStrCpy(m_txn.Payment.d_street_1, pData, dbdatlen(m_dbproc, 9));
			if (pData=dbdata(m_dbproc, 10))

UtilStrCpy(m_txn.Payment.d_street_2, pData, dbdatlen(m_dbproc, 10));
			if (pData=dbdata(m_dbproc, 11))
				UtilStrCpy(m_txn.Payment.d_city,
pData, dbdatlen(m_dbproc, 11));
			if (pData=dbdata(m_dbproc, 12))
				UtilStrCpy(m_txn.Payment.d_state,
pData, dbdatlen(m_dbproc, 12));
			if (pData=dbdata(m_dbproc, 13))
				UtilStrCpy(m_txn.Payment.d_zip,
pData, dbdatlen(m_dbproc, 13));
			if (pData=dbdata(m_dbproc, 14))
				UtilStrCpy(m_txn.Payment.c_first,
pData, dbdatlen(m_dbproc, 14));
			if (pData=dbdata(m_dbproc, 15))

UtilStrCpy(m_txn.Payment.c_middle, pData, dbdatlen(m_dbproc, 15));
			if (pData=dbdata(m_dbproc, 16))

UtilStrCpy(m_txn.Payment.c_street_1, pData, dbdatlen(m_dbproc, 16));
			if (pData=dbdata(m_dbproc, 17))

UtilStrCpy(m_txn.Payment.c_street_2, pData, dbdatlen(m_dbproc, 17));
			if (pData=dbdata(m_dbproc, 18))
				UtilStrCpy(m_txn.Payment.c_city,
pData, dbdatlen(m_dbproc, 18));
			if (pData=dbdata(m_dbproc, 19))
				UtilStrCpy(m_txn.Payment.c_state,
pData, dbdatlen(m_dbproc, 19));
			if (pData=dbdata(m_dbproc, 20))
				UtilStrCpy(m_txn.Payment.c_zip,
pData, dbdatlen(m_dbproc, 20));
			if (pData=dbdata(m_dbproc, 21))

UtilStrCpy(m_txn.Payment.c_phone, pData, dbdatlen(m_dbproc, 21));
			if (pData=dbdata(m_dbproc, 22))
			{
				datetime = *((DBDATETIME *)
pData);
				dbdatecrack(m_dbproc, &daterec,
&datetime);
				m_txn.Payment.c_since.year   =
daterec.year;
				m_txn.Payment.c_since.month  =
daterec.month;
				m_txn.Payment.c_since.day    =
daterec.day;
				m_txn.Payment.c_since.hour   =
daterec.hour;
				m_txn.Payment.c_since.minute =
daterec.minute;
				m_txn.Payment.c_since.second =
daterec.second;
			}
			if(pData=dbdata(m_dbproc, 23))

UtilStrCpy(m_txn.Payment.c_credit, pData, dbdatlen(m_dbproc, 23));
			if(pData=dbdata(m_dbproc, 24))
```

```
                                        dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,24), SQLFLT8, (BYTE
*)&m_txn.Payment.c_credit_lim, 8);
                                if(pData=dbdata(m_dbproc, 25))
                                        dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,25), SQLFLT8, (BYTE
*)&m_txn.Payment.c_discount, 8);
                                if(pData=dbdata(m_dbproc, 26))
                                        dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,26), SQLFLT8, (BYTE
*)&m_txn.Payment.c_balance, 8);
                                if(pData=dbdata(m_dbproc, 27))
                                        UtilStrCpy(m_txn.Payment.c_data,
pData, dbdatlen(m_dbproc, 27));

                                DiscardNextRows(0);
                                DiscardNextResults(0);


                                if (m_txn.Payment.c_id == 0)
                                        throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
                                else
                                        m_txn.Payment.exec_status_code =
eOK;

                                return;
                        }
                        catch (CSQLERR *e)
                        {
                                if ((e->m_msgno == 1205 ||
                                        (e->m_msgno ==
iErrOleDbProvider &&
                                        strstr(e->m_msgtext,
sErrTimeoutExpired) != NULL)) &&
                                        (++iTryCount <= iMaxRetries))
                                {
                                        // hit deadlock; backoff for
increasingly longer period
                                        delete e;
                                        Sleep(10 * iTryCount);
                                }
                                else
                                        throw;
                        }
                }       // while (TRUE)

//              if (iTryCount)
//                      throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}



void CTPCC_DBLIB::OrderStatus()
{
        int                     i;
        DBDATETIME      datetime;
        DBDATEREC       daterec;

        int                     iTryCount = 0;
        RETCODE                 rc;
        const BYTE      *pData;

        ResetError();

        while (TRUE)
        {
                try
                {
                        dbrpcinit(m_dbproc, "tpcc_orderstatus", 0);

                        dbrpcparam(m_dbproc, NULL, 0, SQLINT2,
-1, -1, (BYTE *) &m_txn.OrderStatus.w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1,
-1, -1, (BYTE *) &m_txn.OrderStatus.d_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT4,
-1, -1, (BYTE *) &m_txn.OrderStatus.c_id);

                        // if customer id is zero, then order status is by
name
                        if (m_txn.OrderStatus.c_id == 0)
                                dbrpcparam(m_dbproc, NULL, 0,
SQLCHAR, -1, strlen(m_txn.OrderStatus.c_last), (unsigned char
*)m_txn.OrderStatus.c_last);


                        if (dbrpcexec(m_dbproc) == FAIL)

ThrowError(CDBLIBERR::eDbRpcExec);

                        // Get order lines
                        if (dbresults(m_dbproc) != SUCCEED)
                        {
                                if ((m_DbLibErr == NULL) &&
(m_SqlErr == NULL))
                                        throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
                                else

ThrowError(CDBLIBERR::eDbResults);
                        }

                        if (dbnumcols(m_dbproc) != 5)

ThrowError(CDBLIBERR::eWrongNumCols);

                        i = 0;
                        while (TRUE)
                        {
                                rc = dbnextrow(m_dbproc);
                                if (rc == NO_MORE_ROWS)
                                        break;
                                if (rc != REG_ROW)

ThrowError(CDBLIBERR::eDbNextRow);

                                if(pData=dbdata(m_dbproc, 1))

m_txn.OrderStatus.OL[i].ol_supply_w_id = (*(DBSMALLINT *) pData);
                                if(pData=dbdata(m_dbproc, 2))

m_txn.OrderStatus.OL[i].ol_i_id = (*(DBINT *) pData);
                                if(pData=dbdata(m_dbproc, 3))

m_txn.OrderStatus.OL[i].ol_quantity = (*(DBSMALLINT *) pData);
                                if(pData=dbdata(m_dbproc, 4))
                                        dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,4),

SQLFLT8, (BYTE *)&m_txn.OrderStatus.OL[i].ol_amount, 8);
                                if(pData=dbdata(m_dbproc, 5))
                                {
                                        datetime =
*((DBDATETIME *) pData);
                                        dbdatecrack(m_dbproc,
&daterec, &datetime);

m_txn.OrderStatus.OL[i].ol_delivery_d.year   = daterec.year;

m_txn.OrderStatus.OL[i].ol_delivery_d.month  = daterec.month;
```

```
m_txn.OrderStatus.OL[i].ol_delivery_d.day    = daterec.day;

m_txn.OrderStatus.OL[i].ol_delivery_d.hour   = daterec.hour;

m_txn.OrderStatus.OL[i].ol_delivery_d.minute = daterec.minute;

m_txn.OrderStatus.OL[i].ol_delivery_d.second = daterec.second;
                                        }
                                        i++;
                                }
                                m_txn.OrderStatus.o_ol_cnt = i;


                                if (dbresults(m_dbproc) != SUCCEED)

ThrowError(CDBLIBERR::eDbResults);

                                if (dbnextrow(m_dbproc) != REG_ROW)

ThrowError(CDBLIBERR::eDbNextRow);

                                if (dbnumcols(m_dbproc) != 8)

ThrowError(CDBLIBERR::eWrongNumCols);

                                if(pData=dbdata(m_dbproc, 1))
                                        m_txn.OrderStatus.c_id =
(*(DBINT *) pData);
                                if(pData=dbdata(m_dbproc, 2))

UtilStrCpy(m_txn.OrderStatus.c_last, pData, dbdatlen(m_dbproc,2));
                                if(pData=dbdata(m_dbproc, 3))

UtilStrCpy(m_txn.OrderStatus.c_first, pData, dbdatlen(m_dbproc,3));
                                if(pData=dbdata(m_dbproc, 4))

UtilStrCpy(m_txn.OrderStatus.c_middle, pData, dbdatlen(m_dbproc, 4));

                                if(pData=dbdata(m_dbproc, 5))
                                {
                                        datetime = *((DBDATETIME *)
pData);
                                        dbdatecrack(m_dbproc, &daterec,
&datetime);
                                        m_txn.OrderStatus.o_entry_d.year
= daterec.year;

m_txn.OrderStatus.o_entry_d.month  = daterec.month;
                                        m_txn.OrderStatus.o_entry_d.day
= daterec.day;
                                        m_txn.OrderStatus.o_entry_d.hour
= daterec.hour;

m_txn.OrderStatus.o_entry_d.minute = daterec.minute;

m_txn.OrderStatus.o_entry_d.second = daterec.second;
                                }
                                if(pData=dbdata(m_dbproc, 6))

                                        m_txn.OrderStatus.o_carrier_id =
(*(DBSMALLINT *) pData);
                                if(pData=dbdata(m_dbproc, 7))
                                        dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,7),
                                                                SQLFLT8,
(BYTE *)&m_txn.OrderStatus.c_balance, 8);
                                if(pData=dbdata(m_dbproc, 8))
                                        m_txn.OrderStatus.o_id =
(*(DBINT *) pData);
                                DiscardNextRows(0);
                                DiscardNextResults(0);


                                if (m_txn.OrderStatus.o_ol_cnt == 0)
                                        throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
                                else if (m_txn.OrderStatus.c_id == 0 &&
m_txn.OrderStatus.c_last[0] == 0)
                                        throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
                                else

m_txn.OrderStatus.exec_status_code = eOK;

                                return;
                        }
                        catch (CSQLERR *e)
                        {
                                if ((e->m_msgno == 1205 ||
                                        (e->m_msgno ==
iErrOleDbProvider &&
                                        strstr(e->m_msgtext,
sErrTimeoutExpired) != NULL)) &&
                                        (++iTryCount <= iMaxRetries))
                                {
                                        // hit deadlock; backoff for
increasingly longer period
                                        delete e;
                                        Sleep(10 * iTryCount);
                                }
                                else
                                        throw;
                        }
                }               // while (TRUE)

//              if (iTryCount)
//                      throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}


void CTPCC_DBLIB::Delivery()
{
        int                             i;
        int                             iTryCount = 0;
        const BYTE              *pData;

        ResetError();

        while (TRUE)
        {
                try
                {
                        dbrpcinit(m_dbproc, "tpcc_delivery", 0);

                        dbrpcparam(m_dbproc, NULL, 0, SQLINT2,
-1, -1, (BYTE *) &m_txn.Delivery.w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1,
-1, -1, (BYTE *) &m_txn.Delivery.o_carrier_id);

                        if (dbrpcexec(m_dbproc) == FAIL)

ThrowError(CDBLIBERR::eDbRpcExec);

                        if (dbresults(m_dbproc) != SUCCEED)

ThrowError(CDBLIBERR::eDbResults);
```

```
                        if (dbnextrow(m_dbproc) != REG_ROW)

ThrowError(CDBLIBERR::eDbNextRow);

                        if (dbnumcols(m_dbproc) != 10)

ThrowError(CDBLIBERR::eWrongNumCols);

                        for (i=0; i<10; i++)
                        {
                                if (pData = dbdata(m_dbproc, i+1))
                                        m_txn.Delivery.o_id[i]
= *((DBINT *)pData);
                        }

                        DiscardNextRows(0);
                        DiscardNextResults(0);

                        m_txn.Delivery.exec_status_code = eOK;
                        return;
                }
                catch (CSQLERR *e)
                {
                        if ((e->m_msgno == 1205 ||
                                (e->m_msgno ==
iErrOleDbProvider &&
                                strstr(e->m_msgtext,
sErrTimeoutExpired) != NULL)) &&
                                (++iTryCount <= iMaxRetries))
                        {
                                // hit deadlock; backoff for
increasingly longer period
                                delete e;
                                Sleep(10 * iTryCount);
                        }
                        else
                                throw;
                }
        }       // while (TRUE)

//      if (iTryCount)
//              throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_DBLIB::ResetError()
{
        if (m_DbLibErr != NULL)
        {
                delete m_DbLibErr;
                m_DbLibErr = (CDBLIBERR*)NULL;
        }

        if (m_SqlErr != NULL)
        {
                delete m_SqlErr;
                m_SqlErr = (CSQLERR*)NULL;
        }
        return;
}
```

tpcc_dblib.h

```
/*      FILE:                   TPCC_DBLIB.H
 *                                      Microsoft TPC-C Kit Ver. 4.20.000
 *                                      Copyright Microsoft, 1999
 *      All Rights Reserved
 *                                      Version 4.10.000 audited by
Richard Gimarc, Performance Metrics, 3/17/99
 *
 *      PURPOSE:        Header file for TPC-C txn class
implementation.
 *
 *  Change history:
 *                      4.20.000 - updated rev number to match kit
 */
#pragma once

#ifndef PDBPROCESS
#define DBPROCESS void   // dbprocess structure type
typedef DBPROCESS * PDBPROCESS;
#endif

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

class CSQLERR : public CBaseErr
{
        public:

                CSQLERR(void)
                {
                        m_msgno = 0;
                        m_msgstate = 0;
                        m_severity = 0;
                        m_msgtext = NULL;
                };

                ~CSQLERR()
                {
                        delete [] m_msgtext;
                };

                int             m_msgno;
                int             m_msgstate;
                int             m_severity;
                char   *m_msgtext;

                int ErrorType() {return ERR_TYPE_SQL;};
                int ErrorNum() {return m_msgno;};
                char *ErrorText() {return m_msgtext;};

};

class CDBLIBERR : public CBaseErr
{
        public:
                enum ACTION
                {
                        eNone,
                        eUnknown,
                        eLogin,                         //
error from dblogin
                        eDbOpen,                        // error from
dbopen
                        eDbUse,                         //
error from dbuse
                        eDbSqlExec,                     //
error from dbsqlexec
                        eDbSet,                         //
error from one of the dbset* routines
                        eDbNextRow,                     //
error from dbnextrow
                        eWrongRowCount,                 // more or
less rows returned than expected
```

```
                        eWrongNumCols,          // more or
less columns returned than expected
                        eDbResults,                     //
error from dbresults
                        eDbRpcExec,                     //
error from dbrpcexec
                        eDbSetMaxProcs,         // error from
dbsetmaxprocs
                        eDbProcHandler          // error from
either dbprocerrhandle or dbprocmsghandle
                };

                CDBLIBERR(ACTION eAction, int severity = 0, int
dberror = 0, int oserr = 0)
                {
                        m_eAction = eAction;
                        m_severity = severity;
                        m_dberror = dberror;
                        m_oserr = oserr;

                        m_dberrstr = NULL;
                        m_oserrstr = NULL;
                };

                ~CDBLIBERR()
                {
                        delete [] m_dberrstr;
                        delete [] m_oserrstr;
                };

                ACTION  m_eAction;
                int             m_severity;
                int             m_dberror;
                int             m_oserr;
                char   *m_dberrstr;
                char   *m_oserrstr;

                int ErrorType() {return ERR_TYPE_DBLIB;};
                int ErrorNum() {return m_dberror;};
                char *ErrorText() {return m_dberrstr;};

};

class CTPCC_DBLIB_ERR : public CBaseErr
{
        public:
                enum CTPCC_DBLIB_ERRS
                {
                        ERR_WRONG_SP_VERSION = 1,       //
"Wrong version of stored procs on database server"
                        ERR_INVALID_CUST,
// "Invalid Customer id,name."
                        ERR_NO_SUCH_ORDER,
// "No orders found for customer."
                        ERR_RETRIED_TRANS,
// "Retries before transaction succeeded."
                };

                CTPCC_DBLIB_ERR( int iErr ) { m_errno = iErr;
m_iTryCount = 0; };

                CTPCC_DBLIB_ERR( int iErr, int iTryCount ) { m_errno
= iErr; m_iTryCount = iTryCount; };

                int             m_errno;
                int             m_iTryCount;

                int ErrorType() {return ERR_TYPE_TPCC_DBLIB;};
                int ErrorNum() {return m_errno;};

                char *ErrorText();
};

class DllDecl CTPCC_DBLIB : public CTPCC_BASE
{
        private:
                // declare variables and private functions here...
                PDBPROCESS      m_dbproc;
                CDBLIBERR       *m_DbLibErr;            //
not allocated until needed (maybe never)
                CSQLERR         *m_SqlErr;
// not allocated until needed (maybe never)
                int                     m_MaxRetries;
// retry count on deadlock

                void DiscardNextRows(int iExpectedCount);
                void DiscardNextResults(int iExpectedCount);
                void ThrowError( CDBLIBERR::ACTION eAction );
                void ResetError();

                union
                {
                        NEW_ORDER_DATA
NewOrder;
                        PAYMENT_DATA            Payment;
                        DELIVERY_DATA           Delivery;
                        STOCK_LEVEL_DATA        StockLevel;
                        ORDER_STATUS_DATA       OrderStatus;
                }                       m_txn;

        public:
                CTPCC_DBLIB(LPCSTR szServer, LPCSTR szUser,
LPCSTR szPassword, LPCSTR szHost, LPCSTR szDatabase );
                ~CTPCC_DBLIB(void);

                inline PNEW_ORDER_DATA
BuffAddr_NewOrder()             { return &m_txn.NewOrder;    };
                inline PPAYMENT_DATA
BuffAddr_Payment()      { return &m_txn.Payment;     };
                inline PDELIVERY_DATA
BuffAddr_Delivery()     { return &m_txn.Delivery;    };
                inline PSTOCK_LEVEL_DATA BuffAddr_StockLevel()
{ return &m_txn.StockLevel;  };
                inline PORDER_STATUS_DATA
BuffAddr_OrderStatus()  { return &m_txn.OrderStatus; };

                void NewOrder           ();
                void Payment            ();
                void Delivery           ();
                void StockLevel         ();
                void OrderStatus        ();


                // these are public because they must be called from the
dblib err_handler and msg_hangler
                // outside of the class
                void SetDbLibError(int severity, int dberr, int oserr,
LPCSTR dberrstr, LPCSTR oserrstr);
                void SetSqlError( int msgno, int msgstate, int severity,
LPCSTR msgtext );

};

extern "C" DllDecl CTPCC_DBLIB* CTPCC_DBLIB_new
        ( LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword, LPCSTR
szHost, LPCSTR szDatabase );

typedef CTPCC_DBLIB* (TYPE_CTPCC_DBLIB)(LPCSTR, LPCSTR,
LPCSTR, LPCSTR, LPCSTR);
```

trans.h

```
/*          FILE:                   TRANS.H
 *                                          Microsoft TPC-C Kit Ver. 4.20.000
 *                                          Copyright Microsoft, 1999
 *          All Rights Reserved
 *
 *                                          Version 4.10.000 audited by
Richard Gimarc, Performance Metrics, 3/17/99
 *
 *          PURPOSE:            Header file for TPC-C structure templates.
 *
 *  Change history:
 *                      4.20.000 - updated rev number to match kit
 */
#pragma once

// String length constants
#define SERVER_NAME_LEN         20
#define DATABASE_NAME_LEN       20
#define USER_NAME_LEN           20
#define PASSWORD_LEN            20
#define TABLE_NAME_LEN          20
#define I_DATA_LEN              50
#define I_NAME_LEN              24
#define BRAND_LEN               1
#define LAST_NAME_LEN           16
#define W_NAME_LEN              10
#define ADDRESS_LEN             20
#define STATE_LEN               2
#define ZIP_LEN                 9
#define S_DIST_LEN              24
#define S_DATA_LEN              50
#define D_NAME_LEN              10
#define FIRST_NAME_LEN          16
#define MIDDLE_NAME_LEN         2
#define PHONE_LEN               16
#define DATETIME_LEN            30
#define CREDIT_LEN              2
#define C_DATA_LEN              250
#define H_DATA_LEN              24
#define DIST_INFO_LEN           24
#define MAX_OL_NEW_ORDER_ITEMS  15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN              25
#define OL_DIST_INFO_LEN        24

// TIMESTAMP_STRUCT is provided by the ODBC header file sqltypes.h, but is
not available
// when compiling with dblib, so redefined here.  Note: we are using the symbol
"__SQLTYPES"
// (declared in sqltypes.h) as a way to determine if TIMESTAMP_STRUCT has
been declared.
#ifndef __SQLTYPES
        typedef struct
        {
                short                   /*
SQLSMALLINT */    year;
                unsigned short          /* SQLUSMALLINT */
month;
                unsigned short          /* SQLUSMALLINT */
day;
                unsigned short          /* SQLUSMALLINT */
hour;
                unsigned short          /* SQLUSMALLINT */
minute;
                unsigned short          /* SQLUSMALLINT */
second;
                unsigned long           /* SQLUINTEGER */
fraction;
        } TIMESTAMP_STRUCT;
#endif

// possible values for exec_status_code after transaction completes
enum EXEC_STATUS
{
        eOK,                    // 0        "Transaction
commited."
        eInvalidItem,           // 1        "Item number is not valid."
        eDeliveryFailed         // 2        "Delivery Post Failed."
};

// transaction structures
typedef struct
{
        // input params
        short                           ol_supply_w_id;
        long                            ol_i_id;
        short                           ol_quantity;

        // output params
        char
ol_i_name[I_NAME_LEN+1];
        char
ol_brand_generic[BRAND_LEN+1];
        double                          ol_i_price;
        double                          ol_amount;
        short                           ol_stock;
} OL_NEW_ORDER_DATA;

typedef struct
{
        // input params
        short           w_id;
        short           d_id;
        long            c_id;
        short           o_ol_cnt;

        // output params
        EXEC_STATUS                     exec_status_code;
        char            c_last[LAST_NAME_LEN+1];
        char            c_credit[CREDIT_LEN+1];
        double          c_discount;
        double          w_tax;
        double          d_tax;
        long                    o_id;
        short                           o_commit_flag;
        TIMESTAMP_STRUCT        o_entry_d;
        short           o_all_local;
        double          total_amount;
        OL_NEW_ORDER_DATA
OL[MAX_OL_NEW_ORDER_ITEMS];
} NEW_ORDER_DATA, *PNEW_ORDER_DATA;

typedef struct
{       // input params
        short                   w_id;
        short                   d_id;
        long                    c_id;
        short                   c_d_id;
        short                   c_w_id;
        double                  h_amount;
        char            c_last[LAST_NAME_LEN+1];

        // output params
        EXEC_STATUS                     exec_status_code;
        TIMESTAMP_STRUCT        h_date;
        char                    w_street_1[ADDRESS_LEN+1];
        char                    w_street_2[ADDRESS_LEN+1];
        char                    w_city[ADDRESS_LEN+1];
        char                    w_state[STATE_LEN+1];
```

```
        char                            w_zip[ZIP_LEN+1];
        char                            d_street_1[ADDRESS_LEN+1];
        char                            d_street_2[ADDRESS_LEN+1];
        char                            d_city[ADDRESS_LEN+1];
        char                            d_state[STATE_LEN+1];
        char                    d_zip[ZIP_LEN+1];
        char                            c_first[FIRST_NAME_LEN+1];
        char                            c_middle[MIDDLE_NAME_LEN
+ 1];
        char                            c_street_1[ADDRESS_LEN+1];
        char                            c_street_2[ADDRESS_LEN+1];
        char                            c_city[ADDRESS_LEN+1];
        char                            c_state[STATE_LEN+1];
        char                            c_zip[ZIP_LEN+1];
        char                            c_phone[PHONE_LEN+1];
        TIMESTAMP_STRUCT        c_since;
        char                            c_credit[CREDIT_LEN+1];
        double                          c_credit_lim;
        double                          c_discount;
        double                          c_balance;
        char                            c_data[200+1];
} PAYMENT_DATA, *PPAYMENT_DATA;

typedef struct
{
        long                            ol_i_id;
        short                           ol_supply_w_id;
        short                           ol_quantity;
        double                          ol_amount;
        TIMESTAMP_STRUCT        ol_delivery_d;
} OL_ORDER_STATUS_DATA;

typedef struct
{       // input params
        short           w_id;
        short           d_id;
        long            c_id;
        char            c_last[LAST_NAME_LEN+1];

        // output params
        EXEC_STATUS                             exec_status_code;
        char            c_first[FIRST_NAME_LEN+1];
        char            c_middle[MIDDLE_NAME_LEN+1];
        double           c_balance;
        long            o_id;
        TIMESTAMP_STRUCT        o_entry_d;
        short           o_carrier_id;
        OL_ORDER_STATUS_DATA
OL[MAX_OL_ORDER_STATUS_ITEMS];
        short           o_ol_cnt;
} ORDER_STATUS_DATA, *PORDER_STATUS_DATA;


typedef struct
{       // input params
        short                   w_id;
        short                   o_carrier_id;

        // output params
        EXEC_STATUS                             exec_status_code;
        SYSTEMTIME                      queue_time;
        long                                    o_id[10];               //
id's of delivered orders for districts 1 to 10
} DELIVERY_DATA, *PDELIVERY_DATA;

//This structure is used for posting delivery transactions and for writing them to the
delivery server.
typedef struct _DELIVERY_TRANSACTION
{
        SYSTEMTIME      queue;                          //time
delivery transaction queued
        short           w_id;                           //delivery
warehouse
        short           o_carrier_id;           //carrier id
} DELIVERY_TRANSACTION;

typedef struct
{       // input params
        short                                   w_id;
        short                                   d_id;
        short                                   threshold;

        // output params
        EXEC_STATUS                             exec_status_code;
        long                                    low_stock;
} STOCK_LEVEL_DATA, *PSTOCK_LEVEL_DATA;
```

txnlog.h

```
/*      FILE:           TXNLOG.H
 *                              Microsoft TPC-C Kit Ver. 4.10.000
 *
 *                              NOTE: this file is RTE specific and
should not be included
 *                              in Full Disclosure Reports.
 *
 *                              Copyright Microsoft, 1999
 *
 *      PURPOSE:        Structure definitions for logging delivery txn
completion stats.
 *      Contact:    Charles Levine (clevine@microsoft.com)
 */


  typedef struct _TXN_NEWORDER
        {
    BYTE        OL_Count;       //range 0 to 31
    BYTE        OL_Remote_Count;    //range 0 to 31
    WORD        c_id;
    int         o_id;
        } TXN_NEWORDER;

  typedef struct _TXN_PAYMENT
        {
    BYTE        CustByName;
    BYTE        IsRemote;
        } TXN_PAYMENT;

  typedef struct _TXN_ORDERSTATUS
        {
                BYTE    CustByName;
        } TXN_ORDERSTATUS;

        typedef union _TXN_DETAILS
        {
                TXN_NEWORDER   NewOrder;
                TXN_PAYMENT             Payment;
                TXN_ORDERSTATUS         OrderStatus;
        } TXN_DETAILS;


        // Common header for all records in txn log.  The TxnType field is
        // a switch which identifies the particular variant.
        #define TXN_REC_TYPE_CONTROL            1       //

        #define TXN_REC_TYPE_TPCC                       2
// replaces TRANSACTION_TYPE_TPCC
        #define TXN_REC_TYPE_TPCC_DELIV_DEF     3

        typedef struct _TXN_RECORD_HEADER
```

```
        {
                JULIAN_TIME     TxnStartT0;             // start of txn
                BYTE    TxnType;                        // one of TXN_REC_TYPE_*
                BYTE    TxnSubType;                     // depends on TxnType
        } TXN_RECORD_HEADER, *PTXN_RECORD_HEADER;


        typedef struct _TXN_RECORD_CONTROL
        {
                // common header; must exactly match TXN_RECORD_HEADER
                JULIAN_TIME     TxnStartT0;             // start of txn
                BYTE    TxnType;                        // = TXN_REC_TYPE_CONTROL
                BYTE    TxnSubType;                     // depends on TxnType
                // end of common header

                DWORD   Len;                            // number of bytes after this field
        } TXN_RECORD_CONTROL, *PTXN_RECORD_CONTROL;


        // TPC-C Txn Record Layout:
        //
  //'TxnStartT0' is a Julian timestamp corresponding to the moment the
  //txn is sent to the SUT, i.e., beginning of response time.  Deltas
  //are in milliseconds.  Note that if RTDelay > 0, then the txn was
  //delayed by this amount.  The delay occurs at the beginning of the
  //response time.  So if RTDelay > 0, then the txn was actually sent
  //at TxnStartT0 + RTDelay.
  //
  //Graphically:
  //
  //  time -->
  //
  //    |--- Menu  ---|-- Keying  --|-- Response --|--- Think  --|
  //     <- DeltaT1 -> <- DeltaT2 -> <- DeltaT4  -> <- DeltaT3 ->
  //                     ^
  //                     ^ TxnStartT0
  //
  //RTDelay is the amount of response time delay included in DeltaT4.
  //RTDelay is recorded per txn because this value can be changed on
  //the fly, and so may vary from txn to txn.
  //
  //TxnStatus is the txn completion code.  It is used to indicate errors.
  //For example, in the New Order txn, 1% of txns abort.  TxnStatus will
  //reflect this.

        typedef struct _TXN_RECORD_TPCC
        {
                // common header; must exactly match TXN_RECORD_HEADER
                JULIAN_TIME     TxnStartT0;             // start of txn
                BYTE    TxnType;                        // = TXN_REC_TYPE_TPCC
                BYTE    TxnSubType;                     // depends on TxnType
                // end of common header

                int     DeltaT1;                // menu time (ms)
                int     DeltaT2;                // keying time (ms)
                int     DeltaT3;                // think time (ms)
                int     DeltaT4;                // response time (ms)
                int     RTDelay;                // response time delay (ms)
                int                     TxnError;               // error code providing more detail for TxnStatus
                WORD    w_id;                   // warehouse ID
                BYTE    d_id;                   // assigned district ID for this thread
                BYTE    d_id_ThisTxn;           // district ID chosen for this particular
                BYTE    TxnStatus;              // completion status for txn to indicate errors
                BYTE    reserved;               // for word alignment
                TXN_DETAILS     TxnDetails;             //
        } TXN_RECORD_TPCC, *PTXN_RECORD_TPCC;

        // TPC-C Deferred Delivery Txn Record Layout:
        //
  //Incorporating delivery transaction information into the above
        //structure would increase the size of TXN_DETAILS from 8 to 42 bytes.
  //Hence, we store delivery transaction details in a separate structure.
  //
        typedef struct _TXN_RECORD_TPCC_DELIV_DEF
        {
                // common header; must exactly match TXN_RECORD_HEADER
                JULIAN_TIME     TxnStartT0;             // start of txn
                BYTE    TxnType;                        // = TXN_REC_TYPE_TPCC_DELIV_DEF
                BYTE    TxnSubType;                     // = 0
                // end of common header

                int     DeltaT4;                // response time (ms)
                int                     DeltaTxnExec;           // execution time (ms)
                WORD    w_id;                   // warehouse ID
                BYTE    TxnStatus;              // completion status for txn to indicate errors
                BYTE    reserved;               // for word alignment
                short   o_carrier_id;   // carrier id
                long    o_id[10];               // returned delivery transaction ids
        } TXN_RECORD_TPCC_DELIV_DEF, *PTXN_RECORD_TPCC_DELIV_DEF;



#define         TXN_LOG_VERSION                 1
#define         TXN_DATA_START          4096    // offset in log file where log records start
#define         TXN_LOG_EYE_CATCHER     "BC"    // signature bytes at the start of log file


        ////////////////////////////////////////////////////////////////////
        // The transaction log has a header as the first 4K block.
        //
        typedef struct _TXN_LOG_HEADER
        {
                char                    EyeCatcher[2];          // signature bytes; should always be "BC"
                int                             LogVersion;
        // set to TXN_LOG_VERSION
                JULIAN_TIME             BeginTxnTS;
// timestamp of first (lowest) txn start
                JULIAN_TIME             EndTxnTS;
// timestamp of last (highest) txn completion time
                int                             iRecCount;
// number of records in log file
```

```
                BOOL                    bLogSorted;
                int                     iFileSize;
// file size in bytes

                // the record map provides a fast way to get close to a
particular timestamp in a sorted log file.
//              struct
//              {
//                      JULIAN_TIME             TS;
        // timestamp of record
//                      int
iPos;           // byte position in file
//              }
RecMap[RecMapSize];
//#define       RecMapSize                      200

        } TXN_LOG_HEADER, *PTXN_LOG_HEADER;



#define     READ_BUFFER_SIZE            64*1024
#define     WRITE_BUFFER_SIZE            8*1024

#define     NUM_READ_BUFFERS            1
#define     NUM_WRITE_BUFFERS           2
#define     MAX_NUM_BUFFERS                     2

// flags passed in to the constructor
#define     TXN_LOG_WRITE       0x01
#define     TXN_LOG_READ        0x02
#define     TXN_LOG_SORTED      0x04

#define     TXN_LOG_OS_ERROR    1
#define     TXN_LOG_NOT_SORTED  2

#define SKIP_CTRL_RECS          1

class CTxnLog
{
        private:
                DWORD           iBufferSize;
//buffer allocated size
                DWORD           iBytesFreeInBuffer;
//total bytes available for use in buffer
                int                     iNumBuffers;
        //buffers in use
                int                     iActiveBuffer;
        //indicates which buffer is active: 0 or 1
                int                     iIoBuffer;
        //buffer for any pending IO operation
                int                     iFilePointer;
        //position in file.
                int                     iNextRec;
        //when reading, ordinal value of next record

                // A "save point" is remembered each time GetNextRecord
is called with a start time specified.
                // The next time it is called, if start time is after the save
point, we start scanning from the
                // save point.  This is particularly useful in
FindBestInterval, where the log is scanned repeatedly.
                JULIAN_TIME             SavePtTime;
                int
iSavePtFilePointer;
                int
iSavePtNextRec;

                JULIAN_TIME     lastTS;
        //when writing sorted output, used to verify records are sorted

                BOOL            bWrite;
        //writing log file

                BOOL                    bLogSorted;
        // is log file sorted?  applies to both input and output
                JULIAN_TIME             BeginTxnTS;
        // timestamp of first (lowest) txn start
                JULIAN_TIME             EndTxnTS;
        // timestamp of last (highest) txn completion time
                int                             iRecCount;
        // number of records in log file

                BYTE            *pCurrent;
//ptr to current buffer
                BYTE            *pBuffer[MAX_NUM_BUFFERS];

                PTXN_RECORD_HEADER      *TxnArray;
//transaction record pointer array for sort

                DWORD           dwError;
                HANDLE          hTxnFile;
//handle to log file
                HANDLE          hMapFile;
//map file used when sorting the log
                HANDLE          hIoComplete;
//event to signify that there are no pending IOs
                HANDLE          hLogFileIo;
        //event to signal the IO thread to write the inactive buffer

                Spinlock    Spin;
//spin lock to protect the txn log file buffers

                int Write(BYTE *ptr, DWORD Size);
                static void LogFileIO(CTxnLog *);

        public:
                CTxnLog::CTxnLog(LPCTSTR szFileName, DWORD
dwOpts);
                ~CTxnLog(void);

                int WriteToLog(PTXN_RECORD_TPCC pTxnRcrd);
                int WriteToLog(PTXN_RECORD_TPCC_DELIV_DEF
pTxnRcrd);
                int WriteToLog(PTXN_RECORD_CONTROL pCtrlRec);
                int WriteToLog(PTXN_RECORD_HEADER pCtrlRec);

                int WriteCtrlRecToLog(BYTE SubType, LPTSTR lpStr,
DWORD dwLen);

                void CloseTransactionLogFile(void);

                PTXN_RECORD_HEADER GetNextRecord(BOOL
bSkipCtrlRecs = FALSE);
                PTXN_RECORD_HEADER
GetNextRecord(JULIAN_TIME SeekTimeT0, BOOL bSkipCtrlRecs = FALSE);

                int Sort(void);
                PTXN_RECORD_HEADER GetSortedRecord(int index);

                inline BOOL IsSorted(void) { return bLogSorted; };
                inline JULIAN_TIME BeginTS(void) { return
BeginTxnTS; };
                inline JULIAN_TIME EndTS(void) { return EndTxnTS;
};
                inline int RecordCount(void) { return iRecCount; };
};

class CTXNLOG_ERR : public CBaseErr
{
        public:
```

```
                enum CTPCC_DBLIB_ERRS
                {
                        ERR_BAD_FILE_FORMAT = 1,        // "File format is invalid."
                        ERR_UNKNOWN_LOG_VERSION,        // "Log file version is unknown."
                        ERR_BROKEN_LOG_FILE,            // "Log file is broken."
                        ERR_LOG_NOT_SORTED,             // "Log file is not sorted"
                        ERR_INVALID_TIME_SEQ,           // "Internal Error: Record Time Sequence invalid."
                };

                CTXNLOG_ERR( int iErr ) { m_errno = iErr; };

                int             m_errno;

                int ErrorType() {return ERR_TYPE_TXNLOG;};
                int ErrorNum() {return m_errno;};

                // TODO: need to complete...
                char *ErrorText() {return "";};
};
```

txn_base.h

```
/*      FILE:           TXN_BASE.H
 *                      Microsoft TPC-C Kit Ver. 4.20.000
 *                      Copyright Microsoft, 1999
 *      All Rights Reserved
 *
 *                      Version 4.10.000 audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *      PURPOSE:        Header file for TPC-C txn class implementation.
 *
 *  Change history:
 *              4.20.000 - updated rev number to match kit
 */

#pragma once

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

class DllDecl CTPCC_BASE
{
        public:
                CTPCC_BASE(void) {};
                virtual ~CTPCC_BASE(void) {};

                virtual PNEW_ORDER_DATA BuffAddr_NewOrder()             = 0;
                virtual PPAYMENT_DATA BuffAddr_Payment()        = 0;
                virtual PDELIVERY_DATA BuffAddr_Delivery()      = 0;
                virtual PSTOCK_LEVEL_DATABuffAddr_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;
};
```

webclnt.dsp

```
# Microsoft Developer Studio Project File - Name="webclnt" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 5.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Application" 0x0101

CFG=webclnt - Win32 Release
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "Webclnt.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "Webclnt.mak" CFG="webclnt - Win32 Release"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "webclnt - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "webclnt - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

# Begin Project
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "webclnt - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir ".\Release"
# PROP BASE Intermediate_Dir ".\Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\Release"
# PROP Intermediate_Dir ".\Release"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /machine:I386

!ELSEIF  "$(CFG)" == "webclnt - Win32 Debug"
```

```
# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir ".\Debug"
# PROP BASE Intermediate_Dir ".\Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\Debug"
# PROP Intermediate_Dir ".\Debug"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG"
/D "_WINDOWS" /YX /c
# ADD CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D
"_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib
odbccp32.lib /nologo /subsystem:windows /debug /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386

!ENDIF

# Begin Target

# Name "webclnt - Win32 Release"
# Name "webclnt - Win32 Debug"
# End Target
# End Project
```

webclnt.dsw

```
Microsoft Developer Studio Workspace File, Format Version 6.00
# WARNING: DO NOT EDIT OR DELETE THIS WORKSPACE FILE!

###############################################################################

Project: "db_dblib_dll"=.\db_dblib_dll\db_dblib_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Project: "db_odbc_dll"=.\db_odbc_dll\db_odbc_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Project: "install"=.\install\install.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
   Begin Project Dependency
   Project_Dep_Name isapi_dll
   End Project Dependency
   Begin Project Dependency
   Project_Dep_Name tuxapp
   End Project Dependency
   Begin Project Dependency
   Project_Dep_Name db_dblib_dll
   End Project Dependency
   Begin Project Dependency
   Project_Dep_Name db_odbc_dll
   End Project Dependency
   Begin Project Dependency
   Project_Dep_Name tm_com_dll
   End Project Dependency
   Begin Project Dependency
   Project_Dep_Name tm_tuxedo_dll
   End Project Dependency
   Begin Project Dependency
   Project_Dep_Name tpcc_com_all
   End Project Dependency
   Begin Project Dependency
   Project_Dep_Name tpcc_com_ps
   End Project Dependency
}}}

###############################################################################

Project: "isapi_dll"=.\isapi_dll\isapi_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
   Begin Project Dependency
   Project_Dep_Name db_dblib_dll
   End Project Dependency
   Begin Project Dependency
   Project_Dep_Name db_odbc_dll
   End Project Dependency
   Begin Project Dependency
   Project_Dep_Name tm_tuxedo_dll
   End Project Dependency
   Begin Project Dependency
   Project_Dep_Name tm_com_dll
   End Project Dependency
   Begin Project Dependency
   Project_Dep_Name tm_encina_dll
   End Project Dependency
}}}

###############################################################################

Project: "tm_com_dll"=.\tm_com_dll\tm_com_dll.dsp - Package Owner=<4>
```

```
Package=<5>
{{{
}}}

Package=<4>
{{{
   Begin Project Dependency
   Project_Dep_Name tpcc_com_ps
   End Project Dependency
   Begin Project Dependency
   Project_Dep_Name tpcc_com_all
   End Project Dependency
}}}

###############################################################
###############

Project: "tm_encina_dll"=.\tm_encina_dll\tm_encina_dll.dsp - Package
Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################
###############

Project: "tm_tuxedo_dll"=.\tm_tuxedo_dll\tm_tuxedo_dll.dsp - Package
Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################
###############

Project: "tpcc_com_all"=.\tpcc_com_all\tpcc_com_all.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
   Begin Project Dependency
   Project_Dep_Name tpcc_com_ps
   End Project Dependency
}}}

###############################################################
###############

Project: "tpcc_com_ps"=.\tpcc_com_ps\tpcc_com_ps.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################
###############

Project: "tuxapp"=.\tuxapp\tuxapp.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
   Begin Project Dependency
   Project_Dep_Name db_dblib_dll
   End Project Dependency
   Begin Project Dependency
   Project_Dep_Name db_odbc_dll
   End Project Dependency
}}}

###############################################################
###############

Global:

Package=<5>
{{{
}}}

Package=<3>
{{{
}}}

###############################################################
###############
```

Stored Procedures

neword.sql

payment.sql

ordstat.sql

delivery.sql

stocklev.sql

version.sql

Appendix B: Database Design

backup.sql

```
-- File:    BACKUP.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates backup of tpcc database


declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

dump database tpcc to tpccback1, tpccback2, tpccback3, 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:    BACKUPDEVB.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates tpcc database Backup Devices


use master
go

-- create backup devices

exec sp_addumpdevice 'disk','tpccback1','Y:\tpccback1.dmp'
go
exec sp_addumpdevice 'disk','tpccback2','Y:\tpccback2.dmp'
go
exec sp_addumpdevice 'disk','tpccback3','Z:\tpccback3.dmp'
go
exec sp_addumpdevice 'disk','tpccback4','Z:\tpccback4.dmp'
go
```

createdb.sql

```
-- File:    CREATEDB.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates tpcc database and backup files


use master
go

--          Create temporary table for timing

if exists ( select name from sysobjects where name = 'tpcc_timer' )
        drop table tpcc_timer
go

create table tpcc_timer
(
        start_date                              char(30),
        end_date                                char(30)
)

insert  into tpcc_timer values (0,0)
go

--      Store starting time

update  tpcc_timer
set     start_date  = (select convert(char(30), getdate(),9))
go

-- create main database files

CREATE DATABASE tpcc
ON PRIMARY
(       NAME            = MSSQL70_tpcc_root,
        FILENAME        = "C:\MSSQL70_tpcc_root.mdf",
        SIZE            = 8MB,
        FILEGROWTH      =0),
FILEGROUP       MSSQL70_misc_fg
(       NAME            = MSSQL70_misc1,
        FILENAME        = "F:",
        SIZE            = 18000MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_misc2,
        FILENAME        = "H:",
        SIZE            = 18000MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_misc3,
        FILENAME        = "J:",
        SIZE            = 18000MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_misc4,
        FILENAME        = "L:",
        SIZE            = 18000MB,
        FILEGROWTH      = 0),
FILEGROUP       MSSQL70_cs_fg
(       NAME            = MSSQL70_cs1,
        FILENAME        = "E:",
        SIZE            = 38700MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs2,
        FILENAME        = "G:",
        SIZE            = 38700MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs3,
        FILENAME        = "I:",
        SIZE            = 38700MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs4,
        FILENAME        = "K:",
        SIZE            = 38700MB,
        FILEGROWTH      = 0)
LOG ON
(       NAME            =MSSQL70_tpcc_log,
        FILENAME        ="O:",
        SIZE            =85000MB,
        FILEGROWTH      =0)
go

-- Store ending time
update  tpcc_timer
set     end_date   = (select convert(char(30), getdate(),9))
go

select "Elapsed time (in seconds): ", datediff(second,(select start_date from
tpcc_timer),(select end_date from tpcc_timer))

--      remove temporary table
```

```
if exists ( select name from sysobjects where name = 'tpcc_timer' )
        drop table tpcc_timer
go
```

dbopt1.sql

```
-- File:    DBOPT1.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Sets database options for data load


use master
go

exec sp_dboption tpcc,'select into/bulkcopy',true
exec sp_dboption tpcc,'trunc. log on chkpt.',true
exec sp_dboption tpcc,'torn page detection',false
go

use tpcc
go

checkpoint
go
```

dbopt2.sql

```
-- File:    DBOPT2.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Resets database options after data load

sp_dboption tpcc,'select into/bulkcopy',FALSE
GO

sp_dboption tpcc,'trunc. log on chkpt.',FALSE
GO

USE tpcc
GO

CHECKPOINT
GO

sp_configure 'allow updates',1
GO

RECONFIGURE WITH OVERRIDE
GO

DECLARE         @msg            varchar(50)

IF (SELECT (SUBSTRING((SELECT @@version),1,25))) = 'Microsoft SQL
Server 2000'
        BEGIN
                --                                      --
                --      OPTIONS FOR SQL SERVER 8.0      --
                -- Set option values for user-defined indexes --
                --                                      --

                SET     @msg    = ' '
                PRINT   @msg                            --
                SET     @msg    = 'Setting SQL Server 8.0
indexoptions'
                PRINT   @msg
                SET     @msg    = ' '
                PRINT   @msg                            --

                EXEC sp_indexoption'customer', 'DisAllowPageLocks',
TRUE
                EXEC sp_indexoption'district',  'DisAllowPageLocks',
TRUE
                EXEC sp_indexoption'warehouse',
'DisAllowPageLocks', TRUE
                EXEC sp_indexoption'stock',     'DisAllowPageLocks',
TRUE
                EXEC sp_indexoption'order_line',
'DisAllowRowLocks', TRUE
                EXEC sp_indexoption'orders',    'DisAllowRowLocks',
TRUE
                EXEC sp_indexoption'new_order',
'DisAllowRowLocks', TRUE
                EXEC sp_indexoption'item',
'DisAllowRowLocks', TRUE
                EXEC sp_indexoption'item',
'DisAllowPageLocks', TRUE
        END
ELSE
        BEGIN
                --                                      --
                --      OPTIONS FOR SQL SERVER 7.0      --
                -- Set option values for user-defined indexes --
                --

                SET     @msg    = ' '
                PRINT   @msg                            --
                SET     @msg    = 'Setting SQL Server 7.0
indexoptions'
                PRINT   @msg
                SET     @msg    = ' '
                PRINT   @msg                            --

                EXEC sp_indexoption'customer', 'AllowPageLocks',
FALSE
                EXEC sp_indexoption'district',  'AllowPageLocks',
FALSE
                EXEC sp_indexoption'warehouse',
'AllowPageLocks',       FALSE
                EXEC sp_indexoption'stock',     'AllowPageLocks',
FALSE
                EXEC sp_indexoption'order_line',
'AllowRowLocks',        FALSE
                EXEC sp_indexoption'orders',    'AllowRowLocks',
FALSE
                EXEC sp_indexoption'new_order',
'AllowRowLocks',        FALSE
                EXEC sp_indexoption'item',
'AllowRowLocks',        FALSE
                EXEC sp_indexoption'item',
'AllowPageLocks',       FALSE

        END
GO

Print ' '
Print '********************'
Print 'Pre-specified Locking Hierarchy:'
Print '         Lockflag = 0 ==> No pre-specified hierarchy'
Print '   Lockflag = 1 ==> Lock at Page-level then Table-level'
Print '   Lockflag = 2 ==> Lock at Row-level then Table-level'
Print '   Lockflag = 3 ==> Lock at Table-level'
Print ' '

SELECT   name,lockflags
FROM     sysindexes
WHERE    object_id('warehouse')= id OR
```

```
            object_id('district')     = id OR
            object_id('customer')  = id OR
            object_id('stock')        = id OR
            object_id('orders')       = id OR
            object_id('order_line')= id OR
            object_id('history')      = id OR
            object_id('new_order')= id OR
            object_id('item')         = id
ORDER    BY lockflags asc
GO

sp_configure 'allow updates',0
GO

RECONFIGURE WITH OVERRIDE
GO

EXEC sp_dboption tpcc,           'auto update statistics',FALSE
EXEC sp_dboption tpcc,           'auto create statistics', FALSE
GO

EXEC sp_tableoption 'district',    'pintable',true
EXEC sp_tableoption 'warehouse',           'pintable',true
EXEC sp_tableoption 'new_order',           'pintable',true
EXEC sp_tableoption 'item',                'pintable',true
GO
```

runcfg80.sql

```
/*  TPC-C Benchmark Kit                                          */
/*                                                               */
/*  RUNCFG80.SQL                                                 */
/*                                                               */
/*  This script file is used to set runtime server configuration parameters   */
/*                                                               */


exec sp_configure "show advanced option", 1
go

reconfigure with override
go

/* change this value to approximately the number of connected users */
exec sp_configure "max worker threads",255

/* increase priority of user threads */
exec sp_configure "priority boost",1

/* disable automatic checkpointing */
exec sp_configure "recovery interval",32767

/* change to a mask appropriate for the number of processors on the server */
exec sp_configure "affinity mask",0xf

/* enable fibers */
exec sp_configure "lightweight pooling",1

go

reconfigure with override
go
```

sqlshutdown.sql

```
use tpcc
go
checkpoint
go
shutdown
go
```

verify_msg.sql

```
exec sp_dropmessage 50003
exec sp_addmessage 50003, 1,"Incorrect Sort Order - Please re-install SQL Server
with the Binary Sort Order"
```

verify_sort.sql

```
--  File:    VERIFY_SORT.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.21
--           Copyright Microsoft, 1999, 2000
--  Purpose:  Verifies the Sort Order

if exists (select name from sysobjects where name = "ms_verify_sort" )
             drop procedure ms_verify_sort
go

create proc ms_verify_sort

as

declare      @sort_order             int


-- get the sort order
select       @sort_order             = (select value from sysconfigures where config
= '1123')

if (select @sort_order) <> 50
             RAISERROR (50003,11,1)
go
```

verify_TpccLoad.sql

```
--  File:    VERIFYTPCCLOAD.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.21
--           Copyright Microsoft, 1999, 2000
--  Purpose:  Performs series of TPCC database checks to verify
--                that database load completed correctly

print        " "
select       convert(char(30), getdate(),9)
print        " "

use tpcc
go

--           ************************************
--
--           Check rows per table from SYSINDEXES
--
--           ************************************

print        'WAREHOUSE TABLE'

select       rows
from         sysindexes
where        id             = object_id("warehouse")
go

print        'DISTRICT TABLE = (10 * No of warehouses)'
```

```
select	rows
from	sysindexes
where	id		=object_id("district")
go

print	'ITEM TABLE = 100,000'

select	rows
from	sysindexes
where	id		=object_id("item")
go

print	'CUSTOMER TABLE = (30,000 * No of warehouses)'

select	rows
from	sysindexes
where	id		=object_id("customer")
go

print 'ORDERS TABLE = (30,000 * No of warehouses)'

select	rows
from	sysindexes
where	id		=object_id("orders")
go

print	'HISTORY TABLE = (30,000 * No of warehouses)'

select	rows
from	sysindexes
where	id		=object_id("history")
go

print	'STOCK TABLE = (100,000 * No of warehouses)'

select	rows
from	sysindexes
where	id		=object_id("stock")
go

print	'ORDER_LINE TABLE = (300,000 * No of warehouses + some
change)'

select	rows
from	sysindexes
where	id		=object_id("order_line")
go

print	'NEW_ORDER TABLE = (9000 * No of warehouses)'

select	rows
from	sysindexes
where	id		=object_id("new_order")
go

--		*************
--
--		Check indices
--
--		*************

print '***************Index Check*****************'

use tpcc
go

sp_helpindex		customer
go

sp_helpindex		stock
go

sp_helpindex		district
go

sp_helpindex		item
go

sp_helpindex		new_order
go

sp_helpindex		orders
go

sp_helpindex		order_line
go

sp_helpindex		warehouse
go
```

tables.sql

```
-- File:    TABLES.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates TPC-C tables


use tpcc
go

--
--  Remove all existing TPC-C tables
--

if exists ( select name from sysobjects where name = 'warehouse' )
		drop table warehouse
go
if exists ( select name from sysobjects where name = 'district' )
		drop table district
go
if exists ( select name from sysobjects where name = 'customer' )
		drop table customer
go
if exists ( select name from sysobjects where name = 'history' )
		drop table history
go
if exists ( select name from sysobjects where name = 'new_order' )
		drop table new_order
go
if exists ( select name from sysobjects where name = 'orders' )
		drop table orders
go
if exists ( select name from sysobjects where name = 'order_line' )
		drop table order_line
go
if exists ( select name from sysobjects where name = 'item' )
		drop table item
go
if exists ( select name from sysobjects where name = 'stock' )
		drop table stock
go

--
--  Create new tables
--

create table warehouse
(
		w_id				smallint,
```

```
        w_name                  char(10),
        w_street_1              char(20),
        w_street_2              char(20),
        w_city                  char(20),
        w_state                 char(2),
        w_zip                   char(9),
        w_tax                   numeric(4,4),
        w_ytd                   numeric(12,2)
) on MSSQL70_misc_fg
go

create table district
(
        d_id                    tinyint,
        d_w_id                  smallint,
        d_name                  char(10),
        d_street_1              char(20),
        d_street_2              char(20),
        d_city                  char(20),
        d_state                 char(2),
        d_zip                   char(9),
        d_tax                   numeric(4,4),
        d_ytd                   numeric(12,2),
        d_next_o_id             int
) on MSSQL70_misc_fg
go

create table customer
(
        c_id                    int,
        c_d_id                  tinyint,
        c_w_id                  smallint,
        c_first                 char(16),
        c_middle                char(2),
        c_last                  char(16),
        c_street_1              char(20),
        c_street_2              char(20),
        c_city                  char(20),
        c_state                 char(2),
        c_zip                   char(9),
        c_phone                 char(16),
        c_since                 datetime,
        c_credit                char(2),
        c_credit_lim            numeric(12,2),
        c_discount              numeric(4,4),
        c_balance               numeric(12,2),
        c_ytd_payment           numeric(12,2),
        c_payment_cnt           smallint,
        c_delivery_cnt          smallint,
        c_data                  char(500)
) on MSSQL70_cs_fg
go

create table history
(
        h_c_id                  int,
        h_c_d_id                tinyint,
        h_c_w_id                smallint,
        h_d_id                  tinyint,
        h_w_id                  smallint,
        h_date                  datetime,
        h_amount                numeric(6,2),
        h_data                  char(24)
) on MSSQL70_misc_fg
go

create table new_order
(
        no_o_id                 int,
        no_d_id                 tinyint,
        no_w_id                 smallint
) on MSSQL70_misc_fg
go

create table orders
(
        o_id                    int,
        o_d_id                  tinyint,
        o_w_id                  smallint,
        o_c_id                  int,
        o_entry_d               datetime,
        o_carrier_id            tinyint,
        o_ol_cnt                tinyint,
        o_all_local             tinyint
) on MSSQL70_misc_fg
go

create table order_line
(
        ol_o_id                 int,
        ol_d_id                 tinyint,
        ol_w_id                 smallint,
        ol_number               tinyint,
        ol_i_id                 int,
        ol_supply_w_id          smallint,
        ol_delivery_d           datetime,
        ol_quantity             smallint,
        ol_amount               numeric(6,2),
        ol_dist_info            char(24)
) on MSSQL70_misc_fg
go

create table item
(
        i_id                    int,
        i_im_id                 int,
        i_name                  char(24),
        i_price                 numeric(5,2),
        i_data                  char(50)
) on MSSQL70_misc_fg
go

create table stock
(
        s_i_id                  int,
        s_w_id                  smallint,
        s_quantity              smallint,
        s_dist_01               char(24),
        s_dist_02               char(24),
        s_dist_03               char(24),
        s_dist_04               char(24),
        s_dist_05               char(24),
        s_dist_06               char(24),
        s_dist_07               char(24),
        s_dist_08               char(24),
        s_dist_09               char(24),
        s_dist_10               char(24),
        s_ytd                   int,
        s_order_cnt             smallint,
        s_remote_cnt            smallint,
        s_data                  char(50)
) on MSSQL70_cs_fg
go
```

idxcusnc.sql

```
-- File:    IDXCUSNC.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates non-clustered index on customer table
```

```
use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'customer_nc1' )
        drop index customer.customer_nc1

create unique nonclustered index customer_nc1 on customer(c_w_id, c_d_id,
c_last, c_first, c_id)
        on MSSQL70_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxcuscl.sql

```
-- File:    IDXCUSCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on customer table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'customer_c1' )
        drop index customer.customer_c1

create unique clustered index customer_c1 on customer(c_w_id, c_d_id, c_id)
        on MSSQL70_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxdiscl.sql

```
-- File:    IDXDISCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on district table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'district_c1' )
        drop index district.district_c1

create unique clustered index  district_c1 on district(d_w_id, d_id)
        with fillfactor=100 on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxitmcl.sql

```
-- File:    IDXITMCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on item table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'item_c1' )
        drop index item.item_c1

create unique clustered index item_c1 on item(i_id)
        on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxnodcl.sql

```
-- File:    IDXNODCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on new_order table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'new_order_c1' )
        drop index new_order.new_order_c1

create unique clustered index new_order_c1 on new_order(no_w_id, no_d_id,
no_o_id)
        on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)
```

```
go
```

idxodlcl.sql

```
-- File:    IDXODLCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on order_line table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'order_line_c1' )
            drop index order_line.order_line_c1

create unique clustered index order_line_c1 on order_line(ol_w_id, ol_d_id,
ol_o_id, ol_number)
            on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxordnc.sql

```
-- File:    IDXORDNC.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates non-clustered index on orders table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'orders_nc1' )
            drop index orders.orders_nc1

create index orders_nc1 on orders(o_w_id, o_d_id, o_c_id, o_id)
            on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxordcl.sql

```
-- File:    IDXORDCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on orders table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'orders_c1' )
            drop index orders.orders_c1

create unique clustered index orders_c1 on orders(o_w_id, o_d_id, o_id)
            on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxstkcl.sql

```
-- File:    IDXSTKCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on stock table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'stock_c1' )
            drop index stock.stock_c1

create unique clustered index stock_c1 on stock(s_i_id, s_w_id)
            on MSSQL70_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxwarcl.sql

```
-- File:    IDXWARCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on warehouse table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'warehouse_c1' )
```

```
        drop index warehouse.warehouse_c1

create unique clustered index warehouse_c1 on warehouse(w_id)
        with fillfactor=100 on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

Loader Source Code

getargs.c

```
//      File:           GETARGS.C
//                              Microsoft TPC-C Kit Ver. 4.21
//                              Copyright Microsoft, 1996, 1997, 1998, 1999, 2000
//      Purpose: Source file for command line processing



// Includes
#include "tpcc.h"


//==============================================================================

//
// Function name: GetArgsLoader
//
//==============================================================================


void GetArgsLoader(int argc, char **argv, TPCCLDR_ARGS *pargs)
{
   int          i;
   char   *ptr;

#ifdef DEBUG
   printf("[%ld]DBG: Entering GetArgsLoader()\n", (int) GetCurrentThreadId());
#endif


   /* init args struct with some useful values */
   pargs->server                = SERVER;
   pargs->user                  = USER;
   pargs->password              = PASSWORD;
   pargs->database              = DATABASE;
   pargs->batch                 = BATCH;
   pargs->num_warehouses                = UNDEF;
        pargs->tables_all               = TRUE;
        pargs->table_item                       = FALSE;
        pargs->table_warehouse          = FALSE;
        pargs->table_customer           = FALSE;
        pargs->table_orders                     = FALSE;
        pargs->loader_res_file          = LOADER_RES_FILE;
        pargs->pack_size                = DEFLDPACKSIZE;
        pargs->starting_warehouse       = DEF_STARTING_WAREHOUSE;
        pargs->build_index                      = BUILD_INDEX;
        pargs->index_order                      = INDEX_ORDER;
        pargs->index_script_path  = INDEX_SCRIPT_PATH;
        pargs->scale_down                       = SCALE_DOWN;


   /* check for zero command line args */
   if ( argc == 1 )
                GetArgsLoaderUsage();
```

```
for (i = 1; i < argc; ++i)
{
        if (argv[i][0] != '-' && argv[i][0] != '/')
        {
        printf("\nUnrecognized command");
        GetArgsLoaderUsage();
        exit(1);
        }

        ptr = argv[i];

        switch (ptr[1])
        {
        case 'h':    /* Fall throught */
        case 'H':
                GetArgsLoaderUsage();
                break;

        case 'D':
                pargs->database = ptr+2;
                break;

        case 'P':
                pargs->password = ptr+2;
                break;

        case 'S':
                pargs->server = ptr+2;
                break;

        case 'U':
                pargs->user = ptr+2;
                break;

        case 'b':
                pargs->batch = atol(ptr+2);
                break;

        case 'W':
                pargs->num_warehouses =
atol(ptr+2);
                break;

        case 's':
                pargs->starting_warehouse
= atol(ptr+2);
                break;

        case 't':
                {
                pargs->tables_all
= FALSE;
                if
(strcmp(ptr+2,"item") == 0)
pargs->table_item = TRUE;
                else if
(strcmp(ptr+2,"warehouse") == 0)
pargs->table_warehouse = TRUE;
                else if
(strcmp(ptr+2,"customer") == 0)
pargs->table_customer = TRUE;
                else if
(strcmp(ptr+2,"orders") == 0)
pargs->table_orders = TRUE;
                else
                {
printf("\nUnrecognized command");
```

```
GetArgsLoaderUsage();
                                        exit(1);
                                        }

                                        break;
                                }

                        case 'f':
                                pargs->loader_res_file =
ptr+2;
                                break;

                        case 'p':
                                pargs->pack_size =
atol(ptr+2);
                                break;

                        case 'i':
                                pargs->build_index =
atol(ptr+2);
                                break;

                        case 'o':
                                pargs->index_order =
atol(ptr+2);
                                break;

                        case 'c':
                                pargs->scale_down =
atol(ptr+2);
                                break;

                case 'd':
                                pargs->index_script_path =
ptr+2;
                                break;

                        default:
                                GetArgsLoaderUsage();
                                exit(-1);
                                break;
                        }
        }

        /* check for required args */
        if (pargs->num_warehouses == UNDEF )
        {
                printf("Number of Warehouses is
required\n");
                exit(-2);
        }

        return;
}



//=========================================
============================
//
// Function name: GetArgsLoaderUsage
//
//=========================================
============================

void GetArgsLoaderUsage()
{

#ifdef DEBUG
        printf("[%ld]DBG: Entering GetArgsLoaderUsage()\n", (int)
GetCurrentThreadId());
```

```
#endif



   printf("TPCCLDR:\n\n");

           printf("Parameter
Default\n");


printf("-----------------------------------------------------------------\n");


   printf("-W Number of Warehouses to Load
Required \n");

   printf("-S Server                                %s\n",
SERVER);

   printf("-U Username                               %s\n",
USER);

   printf("-P Password                               %s\n",
PASSWORD);

   printf("-D Database                               %s\n",
DATABASE);

           printf("-b Batch Size
%ld\n", (long) BATCH);

           printf("-p TDS packet size
%ld\n", (long) DEFLDPACKSIZE);

           printf("-f Loader Results Output Filename
 %s\n", LOADER_RES_FILE);

           printf("-s Starting Warehouse
%ld\n", (long) DEF_STARTING_WAREHOUSE);

           printf("-i Build Option (data = 0, data and index = 1)
    %ld\n", (long) BUILD_INDEX);

           printf("-o Cluster Index Build Order (before = 1, after
= 0)  %ld\n", (long) INDEX_ORDER);

           printf("-c Build Scaled Database (normal = 0, tiny =
1)      %ld\n", (long) SCALE_DOWN);

           printf("-d Index Script Path
%s\n", INDEX_SCRIPT_PATH);

           printf("-t Table to Load                          all
tables \n");

   printf("   [item|warehouse|customer|orders]\n");

   printf("   Notes: \n");

   printf("     - the '-t' parameter may be included multiple
times to \n");

   printf("       specify multiple tables to be loaded \n");

   printf("     - 'item' loads ITEM table \n");

   printf("     - 'warehouse' loads WAREHOUSE, DISTRICT,
and STOCK tables \n");

   printf("     - 'customer' loads CUSTOMER and HISTORY
tables \n");

   printf("     - 'orders' load NEW-ORDER, ORDERS,
ORDER-LINE tables \n");



           printf("\nNote:  Command line switches are case
sensitive.\n");



   exit(0);

}
```

strings.c

```
//              File:                   STRINGS.C
//                                              Microsoft TPC-C Kit Ver. 4.21
//                                              Copyright Microsoft, 1996, 1997,
1998, 1999, 2000
//              Purpose:    Source file for database loader string functions


// Includes
#include "tpcc.h"
#include <string.h>
#include <ctype.h>

//=========================================================
================
//
// Function name: MakeAddress
//
//=========================================================
================

void MakeAddress(char *street_1,
                                                char *street_2,
                                                char *city,
                                                char *state,
                                                char *zip)
{

#ifdef DEBUG
   printf("[%ld]DBG: Entering MakeAddress()\n", (int) GetCurrentThreadId());
#endif

   MakeAlphaString (10, 20, ADDRESS_LEN, street_1);
   MakeAlphaString (10, 20, ADDRESS_LEN, street_2);
   MakeAlphaString (10, 20, ADDRESS_LEN, city);
   MakeAlphaString ( 2,  2, STATE_LEN, state);
   MakeZipNumberString( 9,  9, ZIP_LEN, zip);
```

```
#ifdef DEBUG
   printf("[%ld]DBG: MakeAddress: street_1: %s, street_2: %s, city: %s, state:
%s, zip: %s\n",
                                 (int) GetCurrentThreadId(), street_1, street_2,
city, state, zip);
#endif

   return;

}


//=================================================================
================
//
// Function name: LastName
//
//=================================================================
================

void LastName(int  num,
                        char *name)
{
   static char *n[] =
   {
                        "BAR" , "OUGHT", "ABLE" , "PRI"  , "PRES",
                        "ESE" , "ANTI" , "CALLY", "ATION", "EING"
   };

#ifdef DEBUG
   printf("[%ld]DBG: Entering LastName()\n", (int) GetCurrentThreadId());
#endif

   if ((num >= 0) && (num < 1000))
   {
                        strcpy(name, n[(num/100)%10]);
                        strcat(name, n[(num/10)%10]);
                        strcat(name, n[(num/1)%10]);

                        if (strlen(name) < LAST_NAME_LEN)
                        {
                                PaddString(LAST_NAME_LEN, name);
                        }

   }
   else
   {
                        printf("\nError in LastName()... num <%ld> out of range
(0,999)\n", num);
                        exit(-1);
   }


#ifdef DEBUG
   printf("[%ld]DBG: LastName: num = [%d] ==> [%d][%d][%d]\n",
                                 (int) GetCurrentThreadId(), num, num/100,
(num/10)%10, num%10);
            printf("[%ld]DBG: LastName: String = %s\n", (int)
GetCurrentThreadId(), name);
#endif

   return;

}


//=================================================================
================
//
// Function name: MakeAlphaString
//
//=================================================================
================

//philipdu 08/13/96 Changed MakeAlphaString to use A-Z, a-z, and 0-9 in
//accordance with spec see below:
//The spec says:
//4.3.2.2     The notation random a-string [x .. y]
//(respectively, n-string [x .. y]) represents a string of random alphanumeric
//(respectively, numeric) characters of a random length of minimum x, maximum
y,
//and mean (y+x)/2. Alphanumerics are A..Z, a..z, and 0..9.  The only other
//requirement is that the character set used "must be able to represent a minimum
//of 128 different characters".  We are using 8-bit chars, so this is a non issue.
//It is completely unreasonable to stuff non-printing chars into the text fields.
//-CLevine 08/13/96

int MakeAlphaString( int  x, int  y, int  z, char *str)
{
            int                 len;
            int                 i;
   char    cc = 'a';
            static      char chArray[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuv
wxyz";
            static      int         chArrayMax = 61;

#ifdef DEBUG
   printf("[%ld]DBG: Entering MakeAlphaString()\n", (int)
GetCurrentThreadId());
#endif

            len= RandomNumber(x, y);

            for (i=0; i<len; i++)
            {
                        cc = chArray[RandomNumber(0, chArrayMax)];
      str[i] = cc;
   }
            if ( len < z )
                        memset(str+len, ' ', z - len);
            str[len] = 0;

   return len;
}


//=================================================================
================
//
// Function name: MakeOriginalAlphaString
//
//=================================================================
================

int MakeOriginalAlphaString(int  x,
                                                                  int  y,

int  z,
                                                                  char *str,
                                                                int  percent)
{
   int                  len;
   int                  val;
   int                  start;

#ifdef DEBUG
   printf("[%ld]DBG: Entering MakeOriginalAlphaString()\n", (int)
GetCurrentThreadId());
#endif
```

```
   // verify prercentage is valid
   if ((percent < 0) || (percent > 100))
   {
                        printf("MakeOrigianlAlphaString: Invalid percentage:
%d\n", percent);
                        exit(-1);
   }

   // verify string is at least 8 chars in length
   if ((x + y) <= 8)
   {
                        printf("MakeOriginalAlphaString: string length must be >=
8\n");
                        exit(-1);
   }

   // Make Alpha String
   len = MakeAlphaString(x,y, z, str);

   val = RandomNumber(1,100);
   if (val <= percent)
   {
                        start = RandomNumber(0, len - 8);
                        strncpy(str + start, "ORIGINAL", 8);
   }


#ifdef DEBUG
   printf("[%ld]DBG: MakeOriginalAlphaString: : %s\n",
                                (int) GetCurrentThreadId(), str);
#endif

   return strlen(str);
}


//=====================================================
================
//
// Function name: MakeNumberString
//
//=====================================================
================
int MakeNumberString(int  x, int  y, int  z, char *str)
{
              char tmp[16];

              //MakeNumberString is always called MakeZipNumberString(16, 16,
16, string)

              memset(str, '0', 16);
              itoa(RandomNumber(0, 99999999), tmp, 10);
              memcpy(str, tmp, strlen(tmp));

              itoa(RandomNumber(0, 99999999), tmp, 10);
              memcpy(str+8, tmp, strlen(tmp));

              str[16] = 0;

   return 16;
}


//=====================================================
================
//
// Function name: MakeZipNumberString
//
//=====================================================
================
int MakeZipNumberString(int  x, int  y, int  z, char *str)
{
              char tmp[16];

              //MakeZipNumberString is always called MakeZipNumberString(9, 9,
9, string)

              strcpy(str, "000011111");

              itoa(RandomNumber(0, 9999), tmp, 10);
              memcpy(str, tmp, strlen(tmp));

   return 9;
}


//=====================================================
================
//
// Function name: InitString
//
//=====================================================
================
void InitString(char *str, int len)
{
#ifdef DEBUG
   printf("[%ld]DBG: Entering InitString()\n", (int) GetCurrentThreadId());
#endif

              memset(str, ' ', len);
   str[len] = 0;
}


//=====================================================
================
// Function name: InitAddress
//
// Description:
//
//=====================================================
================

void InitAddress(char *street_1, char *street_2, char *city, char *state, char *zip)
{
              memset(street_1, ' ', ADDRESS_LEN+1);
              memset(street_2, ' ', ADDRESS_LEN+1);
              memset(city, ' ', ADDRESS_LEN+1);

   street_1[ADDRESS_LEN+1] = 0;
   street_2[ADDRESS_LEN+1] = 0;
   city[ADDRESS_LEN+1]     = 0;

              memset(state, ' ', STATE_LEN+1);
   state[STATE_LEN+1] = 0;

              memset(zip, ' ', ZIP_LEN+1);
   zip[ZIP_LEN+1] = 0;
}


//=====================================================
================
//
// Function name: PaddString
//
//=====================================================
================

void PaddString(int max, char *name)
{
              int                   len;
```

```
        len = strlen(name);
        if ( len < max )
                memset(name+len, ' ', max - len);
        name[max] = 0;

        return;
}
```

tpcc.h

```
//      File:           TPCC.H
//                              Microsoft TPC-C Kit Ver. 4.21
//                              Copyright Microsoft, 1996, 1997,
1998, 1999, 2000
//      Purpose:    Header file for TPC-C database loader


// Build number of TPC Benchmark Kit
#define TPCKIT_VER          "4.21"

// General headers
#include <windows.h>
#include <winbase.h>
#include <stdlib.h>
#include <stdio.h>
#include <process.h>
#include <stddef.h>
#include <stdarg.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <sys\types.h>

// ODBC headers
#include <sql.h>
#include <sqlext.h>
#include <odbcss.h>

// General constants
#define MILLI               1000
#define FALSE                       0
#define TRUE                        1
#define UNDEF                                               -1
#define MINPRINTASCII                       32
#define MAXPRINTASCII                       126

// Default environment constants
#define SERVER                              ""
#define DATABASE                            "tpcc"
#define USER                        "sa"
#define PASSWORD                      ""

// Default loader arguments
#define BATCH                       10000
#define DEFLDPACKSIZE                       32768
#define LOADER_RES_FILE              "logs\\load.out"
#define LOADER_NURAND_C                     123
#define DEF_STARTING_WAREHOUSE              1
#define BUILD_INDEX                                         1
// build both data and indexes
#define INDEX_ORDER                                         1
// build indexes before load
#define SCALE_DOWN                                          0
  // build a normal scale database
#define INDEX_SCRIPT_PATH "scripts"

typedef struct
{
   char                             *server;
   char                             *database;
   char                             *user;
   char                             *password;
        BOOL                                tables_all;
// set if loading all tables
        BOOL                                table_item;
// set if loading ITEM table specifically
        BOOL                                table_warehouse;    //
set if loading WAREHOUSE, DISTRICT, and STOCK
        BOOL                                table_customer;
// set if loading CUSTOMER and HISTORY
        BOOL                                table_orders;
// set if loading NEW-ORDER, ORDERS, ORDER-LINE
   long                             num_warehouses;
   long                             batch;
   long                             verbose;
        long                                pack_size;
        char                                *loader_res_file;
        char                                *synch_servername;
        long                                case_sensitivity;
        long                                starting_warehouse;
        long                                build_index;
        long                                index_order;
        long                                scale_down;
        char                                *index_script_path;
} TPCCLDR_ARGS;

// String length constants
        #define SERVER_NAME_LEN         20
        #define DATABASE_NAME_LEN       20
        #define USER_NAME_LEN           20
        #define PASSWORD_LEN            20
        #define TABLE_NAME_LEN          20
        #define I_DATA_LEN              50
        #define I_NAME_LEN              24
        #define BRAND_LEN               1
        #define LAST_NAME_LEN           16
        #define W_NAME_LEN              10
        #define ADDRESS_LEN             20
        #define STATE_LEN               2
        #define ZIP_LEN                 9
        #define S_DIST_LEN              24
        #define S_DATA_LEN              50
        #define D_NAME_LEN              10
        #define FIRST_NAME_LEN          16
        #define MIDDLE_NAME_LEN         2
        #define PHONE_LEN               16
        #define CREDIT_LEN              2
        #define C_DATA_LEN              500
        #define H_DATA_LEN              24
        #define DIST_INFO_LEN           24
        #define MAX_OL_NEW_ORDER_ITEMS  15
        #define MAX_OL_ORDER_STATUS_ITEMS 15
        #define STATUS_LEN              25
        #define OL_DIST_INFO_LEN                24
        #define C_SINCE_LEN
23
        #define H_DATE_LEN
23
        #define OL_DELIVERY_D_LEN                           23
        #define O_ENTRY_D_LEN           23


// Functions in random.c
void        seed();
long        irand();
double      drand();
void        WUCreate();
short       WURand();
long        RandomNumber(long lower, long upper);

// Functions in getargs.c;
void        GetArgsLoader();
```

```
void        GetArgsLoaderUsage();

// Functions in time.c
long        TimeNow();

// Functions in strings.c
void        MakeAddress();
void        LastName();
int         MakeAlphaString();
int         MakeOriginalAlphaString();
int         MakeNumberString();
int         MakeZipNumberString();
void        InitString();
void        InitAddress();
void        PaddString();
```

tpccldr.c

```
//          File:                   TPCCLDR.C
//                                          Microsoft TPC-C Kit Ver. 4.21
//                                          Copyright Microsoft, 1996, 1997,
1998, 1999, 2000
//          Purpose:    Source file for TPC-C database loader


// Includes
#include "tpcc.h"
#include "search.h"

// Defines
#define MAXITEMS            100000
#define MAXITEMS_SCALE_DOWN                 100
#define CUSTOMERS_PER_DISTRICT  3000
#define CUSTOMERS_SCALE_DOWN    30
#define DISTRICT_PER_WAREHOUSE  10
#define ORDERS_PER_DISTRICT     3000
#define ORDERS_SCALE_DOWN       30
#define MAX_CUSTOMER_THREADS    2
#define MAX_ORDER_THREADS       3
#define MAX_MAIN_THREADS        4

// Functions declarations

void HandleErrorDBC (SQLHDBC  hdbc1);

void CheckSQL();
void CheckDataBase();

long NURand();
void LoadItem();
void LoadWarehouse();

void Stock();
void District();

void LoadCustomer();
void CustomerBufInit();
void CustomerBufLoad();
void LoadCustomerTable();
void LoadHistoryTable();

void LoadOrders();
void OrdersBufInit();
void OrdersBufLoad();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void GetPermutation();
void CheckForCommit();
void OpenConnections();
void BuildIndex();
void FormatDate ();

// Shared memory structures

typedef struct
{
   long          ol;
   long          ol_i_id;
   short         ol_supply_w_id;
   short         ol_quantity;
   double          ol_amount;
   char          ol_dist_info[DIST_INFO_LEN+1];
            char
ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCT;

typedef struct
{
   long          o_id;
   short         o_d_id;
   short         o_w_id;
   long          o_c_id;
   short         o_carrier_id;
   short         o_ol_cnt;
   short         o_all_local;
            ORDER_LINE_STRUCT   o_ol[15];
} ORDERS_STRUCT;

typedef struct
{
   long                                 c_id;
   short                                c_d_id;
   short                                c_w_id;
            char
c_first[FIRST_NAME_LEN+1];
   char
c_middle[MIDDLE_NAME_LEN+1];
   char                                 c_last[LAST_NAME_LEN+1];
   char                                 c_street_1[ADDRESS_LEN+1];
   char                                 c_street_2[ADDRESS_LEN+1];
   char                                 c_city[ADDRESS_LEN+1];
   char                                 c_state[STATE_LEN+1];
   char                                 c_zip[ZIP_LEN+1];
   char                                 c_phone[PHONE_LEN+1];
   char                                 c_credit[CREDIT_LEN+1];
   double                               c_credit_lim;
   double                               c_discount;
//  fix to avoid ODBC float to numeric conversion problem.
//  double                               c_balance;
            char                                c_balance[6];

   double                               c_ytd_payment;
            short                               c_payment_cnt;
            short                               c_delivery_cnt;
            char
c_data[C_DATA_LEN+1];
            double                              h_amount;
   char                                 h_data[H_DATA_LEN+1];
} CUSTOMER_STRUCT;

typedef struct
{
            char
c_last[LAST_NAME_LEN+1];
            char
c_first[FIRST_NAME_LEN+1];
   long                                 c_id;
} CUSTOMER_SORT_STRUCT;
```

```
typedef struct
{
   long              time_start;
} LOADER_TIME_STRUCT;


// Global variables

char        szLastError[300];

HENV      henv;

HDBC      v_hdbc;                                                    //
for SQL Server version verification
HDBC      i_hdbc1;                                                   // for ITEM
table
HDBC      w_hdbc1;                                                   // for
WAREHOUSE, DISTRICT, STOCK
HDBC      c_hdbc1;                                                   // for
CUSTOMER
HDBC      c_hdbc2;                                                   // for
HISTORY
HDBC      o_hdbc1;                                                   // for
ORDERS
HDBC      o_hdbc2;                                                   // for
NEW-ORDER
HDBC      o_hdbc3;                                                   // for
ORDER-LINE

HSTMT     v_hstmt;                                                   // for SQL
Server version verification
HSTMT     i_hstmt1;
HSTMT     w_hstmt1;
HSTMT     c_hstmt1, c_hstmt2;
HSTMT     o_hstmt1, o_hstmt2, o_hstmt3;

ORDERS_STRUCT   orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT customer_buf[CUSTOMERS_PER_DISTRICT];
long          orders_rows_loaded;
long          new_order_rows_loaded;
long          order_line_rows_loaded;
long          history_rows_loaded;
long          customer_rows_loaded;
long          stock_rows_loaded;
long          district_rows_loaded;
long          item_rows_loaded;
long          warehouse_rows_loaded;
long          main_time_start;
long          main_time_end;
long                              max_items;
long                              customers_per_district;
long                              orders_per_district;
long                              first_new_order;
long                              last_new_order;


TPCCLDR_ARGS    *aptr, args;


//=================================================================
================
//
// Function name: main
//
//=================================================================
================

int main(int  argc, char **argv)
{

            DWORD         dwThreadID[MAX_MAIN_THREADS];
            HANDLE        hThread[MAX_MAIN_THREADS];
            FILE          *fLoader;
            char          buffer[255];
            int                                       i;

            for (i=0; i<MAX_MAIN_THREADS; i++)
                        hThread[i] = NULL;

   printf("\n*************************************************");
            printf("\n*                                       *");
            printf("\n*  Microsoft SQL Server                 *");
            printf("\n*                                       *");
            printf("\n*  TPC-C BENCHMARK KIT:  Database loader        *");
            printf("\n*  Version %s                            *", TPCKIT_VER);
            printf("\n*                                       *");

printf("\n*************************************************\n\n");

   // process command line arguments

   aptr = &args;
   GetArgsLoader(argc, argv, aptr);

            // verify database and tables exist before attempting to load

            CheckDataBase();

            printf("Build interface is ODBC.\n");

            if (aptr->build_index == 0)
                        printf("Data load only - no index creation.\n");
            else
                        printf("Data load and index creation.\n");

            if (aptr->index_order == 0)
                        printf("Clustered indexes will be created after bulk
load.\n");
            else
                        printf("Clustered indexes will be created before bulk
load.\n");

            // set database scale values
            if (aptr->scale_down == 1)
            {
                        printf("*** Scaled Down Database ***\n");
                        max_items = MAXITEMS_SCALE_DOWN;
                        customers_per_district =
CUSTOMERS_SCALE_DOWN;
                        orders_per_district = ORDERS_SCALE_DOWN;
                        first_new_order = 0;
                        last_new_order  = 30;
            }
            else
            {
                        max_items = MAXITEMS;
                        customers_per_district =
CUSTOMERS_PER_DISTRICT;
                        orders_per_district = ORDERS_PER_DISTRICT;
                        first_new_order = 2100;
                        last_new_order  = 3000;
            }


            // open connections to SQL Server

            OpenConnections();

            // open file for loader results
   fLoader = fopen(aptr->loader_res_file, "w");

   if (fLoader == NULL)
```

```
        {
                printf("Error, loader result file open failed.");
                exit(-1);
        }

   // start loading data

   sprintf(buffer,"TPC-C load started for %ld
warehouses.\n",aptr->num_warehouses);

        printf("%s",buffer);
        fprintf(fLoader,"%s",buffer);

        main_time_start = (TimeNow() / MILLI);

        // start parallel load threads


        if (aptr->tables_all || aptr->table_item)
        {
                fprintf(fLoader, "\nStarting loader threads for: item\n");

                hThread[0] = CreateThread(NULL,

 0,

 (LPTHREAD_START_ROUTINE) LoadItem,

 NULL,

 0,

 &dwThreadID[0]);

                if (hThread[0] == NULL)
                {
                        printf("Error, failed in creating creating thread
= 0.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_warehouse)
        {
                fprintf(fLoader, "Starting loader threads for:
warehouse\n");

                hThread[1] = CreateThread(NULL,

 0,

 (LPTHREAD_START_ROUTINE) LoadWarehouse,

 NULL,

 0,

 &dwThreadID[1]);

                if (hThread[1] == NULL)
                {
                        printf("Error, failed in creating creating thread
= 1.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_customer)
        {
                fprintf(fLoader, "Starting loader threads for: customer\n");

                hThread[2] = CreateThread(NULL,

 0,

 (LPTHREAD_START_ROUTINE) LoadCustomer,

 NULL,

 0,

 &dwThreadID[2]);

                if (hThread[2] == NULL)
                {
                        printf("Error, failed in creating creating main
thread = 2.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_orders)
        {
                fprintf(fLoader, "Starting loader threads for: orders\n");

                hThread[3] = CreateThread(NULL,

 0,

 (LPTHREAD_START_ROUTINE) LoadOrders,

 NULL,

 0,

 &dwThreadID[3]);

                if (hThread[3] == NULL)
                {
                        printf("Error, failed in creating creating main
thread = 3.\n");
                        exit(-1);
                }
        }

        // Wait for threads to finish...
        for (i=0; i<MAX_MAIN_THREADS; i++)
        {
                if (hThread[i] != NULL)
                {
                        WaitForSingleObject( hThread[i], INFINITE );

                        CloseHandle(hThread[i]);
                        hThread[i] = NULL;
                }
        }

        main_time_end = (TimeNow() / MILLI);

   sprintf(buffer,"\nTPC-C load completed successfully in %ld minutes.\n",
                        (main_time_end - main_time_start)/60);

        printf("%s",buffer);
        fprintf(fLoader, "%s", buffer);

        fclose(fLoader);

        SQLFreeEnv(henv);

   exit(0);

        return 0;
```

```
}


//=========================================================
================
//
// Function name: LoadItem
//
//=========================================================
================

void LoadItem()
{
   long              i_id;
        long              i_im_id;
   char              i_name[I_NAME_LEN+1];
   double            i_price;
   char              i_data[I_DATA_LEN+1];
        char              name[20];
        long              time_start;
        RETCODE                   rc;
        DBINT             rcint;
        char              bcphint[128];

        // Seed with unique number
        seed(1);

        printf("Loading item table...\n");

        // if build index before load
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
                BuildIndex("idxitmcl");

        InitString(i_name, I_NAME_LEN+1);
        InitString(i_data, I_DATA_LEN+1);

        sprintf(name, "%s..%s", aptr->database, "item");

        rc = bcp_init(i_hdbc1, name, NULL, "logs\\item.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (i_id),
ROWS_PER_BATCH = 100000");
                rc = bcp_control(i_hdbc1, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);
        }

        rc = bcp_bind(i_hdbc1, (BYTE *) &i_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        rc = bcp_bind(i_hdbc1, (BYTE *) i_name, 0, I_NAME_LEN, NULL,
0, 0, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        rc = bcp_bind(i_hdbc1, (BYTE *) &i_price, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0, I_DATA_LEN, NULL,
0, 0, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        time_start = (TimeNow() / MILLI);

        item_rows_loaded = 0;

        for (i_id = 1; i_id <= max_items; i_id++)
        {
                i_im_id = RandomNumber(1L, 10000L);

                MakeAlphaString(14, 24, I_NAME_LEN, i_name);

                i_price = ((float) RandomNumber(100L, 10000L))/100.0;

                MakeOriginalAlphaString(26, 50, I_DATA_LEN, i_data,
10);

                rc = bcp_sendrow(i_hdbc1);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

                item_rows_loaded++;
                CheckForCommit(i_hdbc1, i_hstmt1, item_rows_loaded,
"item", &time_start);
        }

        rcint = bcp_done(i_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(i_hdbc1);

        printf("Finished loading item table.\n");

        SQLFreeStmt(i_hstmt1, SQL_DROP);
        SQLDisconnect(i_hdbc1);
        SQLFreeConnect(i_hdbc1);

        // if build index after load
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxitmcl");

}


//=========================================================
======================
//
// Function   : LoadWarehouse
//
// Loads WAREHOUSE table and loads Stock and District as Warehouses are
created
//
//=========================================================
======================

void LoadWarehouse()
{
   short     w_id;
   char      w_name[W_NAME_LEN+1];
   char      w_street_1[ADDRESS_LEN+1];
   char      w_street_2[ADDRESS_LEN+1];
   char      w_city[ADDRESS_LEN+1];
   char      w_state[STATE_LEN+1];
   char      w_zip[ZIP_LEN+1];
   double    w_tax;
   double    w_ytd;
        char      name[20];
        long      time_start;
        RETCODE           rc;
```

```
	DBINT	rcint;
	char	bcphint[128];


	// Seed with unique number
	seed(2);

	printf("Loading warehouse table...\n");

	// if build index before load...
	if ((aptr->build_index == 1) && (aptr->index_order == 1))
		BuildIndex("idxwarcl");

	InitString(w_name, W_NAME_LEN+1);
	InitAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

	sprintf(name, "%s..%s", aptr->database, "warehouse");

	rc = bcp_init(w_hdbc1, name, NULL, "logs\\whouse.err", DB_IN);

	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	if ((aptr->build_index == 1) && (aptr->index_order == 1))
	{
		sprintf(bcphint, "tablock, order (w_id),
ROWS_PER_BATCH = %d", aptr->num_warehouses);
		rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
		if (rc != SUCCEED)
			HandleErrorDBC(w_hdbc1);
	}

	rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 1);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0, W_NAME_LEN,
NULL, 0, 0, 2);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) w_street_1, 0, ADDRESS_LEN,
NULL, 0, 0, 3);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) w_street_2, 0, ADDRESS_LEN,
NULL, 0, 0, 4);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) w_city, 0, ADDRESS_LEN,
NULL, 0, 0, 5);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) w_state, 0, STATE_LEN, NULL,
0, 0, 6);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) w_zip, 0, ZIP_LEN, NULL, 0, 0,
7);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 8);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 9);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	time_start = (TimeNow() / MILLI);

	warehouse_rows_loaded = 0;

	for (w_id = (short)aptr->starting_warehouse; w_id <=
aptr->num_warehouses; w_id++)
	{
		MakeAlphaString(6,10, W_NAME_LEN, w_name);

		MakeAddress(w_street_1, w_street_2, w_city, w_state,
w_zip);

		w_tax = ((float) RandomNumber(0L,2000L))/10000.00;

		w_ytd = 300000.00;

		rc = bcp_sendrow(w_hdbc1);
		if (rc != SUCCEED)
			HandleErrorDBC(w_hdbc1);

		warehouse_rows_loaded++;
		CheckForCommit(w_hdbc1, i_hstmt1,
warehouse_rows_loaded, "warehouse", &time_start);
	}

	rcint = bcp_done(w_hdbc1);
	if (rcint < 0)
		HandleErrorDBC(w_hdbc1);

	printf("Finished loading warehouse table.\n");

	// if build index after load...
	if ((aptr->build_index == 1) && (aptr->index_order == 0))
		BuildIndex("idxwarcl");

	stock_rows_loaded = 0;
	district_rows_loaded = 0;

	District();
	Stock();

}


//=================================================
================
//
// Function   : District
//
//=================================================
================

void District()
{
   short	d_id;
   short	d_w_id;
   char	d_name[D_NAME_LEN+1];
   char	d_street_1[ADDRESS_LEN+1];
   char	d_street_2[ADDRESS_LEN+1];
   char	d_city[ADDRESS_LEN+1];
   char	d_state[STATE_LEN+1];
   char	d_zip[ZIP_LEN+1];
   double	d_tax;
   double	d_ytd;
	char	name[20];
```

```
    long    d_next_o_id;
            long        time_start;
            int                 w_id;
            RETCODE             rc;
            DBINT       rcint;
            char        bcphint[128];

            // Seed with unique number
            seed(4);

            printf("Loading district table...\n");

            // build index before load
            if ((aptr->build_index == 1) && (aptr->index_order == 1))
                        BuildIndex("idxdiscl");

            InitString(d_name, D_NAME_LEN+1);
            InitAddress(d_street_1, d_street_2, d_city, d_state, d_zip);
            sprintf(name, "%s..%s", aptr->database, "district");

            rc = bcp_init(w_hdbc1, name, NULL, "logs\\district.err", DB_IN);

            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            if ((aptr->build_index == 1) && (aptr->index_order == 1))
            {
                        sprintf(bcphint, "tablock, order (d_w_id, d_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 10));
                        rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
                        if (rc != SUCCEED)
                                    HandleErrorDBC(w_hdbc1);
            }

            rc = bcp_bind(w_hdbc1, (BYTE *) &d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 1);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) &d_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) d_name, 0, D_NAME_LEN,
NULL, 0, 0, 3);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) d_street_1, 0, ADDRESS_LEN,
NULL, 0, 0, 4);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) d_street_2, 0, ADDRESS_LEN,
NULL, 0, 0, 5);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) d_city, 0, ADDRESS_LEN,
NULL, 0, 0, 6);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) d_state, 0, STATE_LEN, NULL,
0, 0, 7);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) d_zip, 0, ZIP_LEN, NULL, 0, 0,
8);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) &d_tax, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 9);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) &d_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 10);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) &d_next_o_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 11);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            d_ytd  = 30000.0;

            d_next_o_id = orders_per_district+1;

            time_start = (TimeNow() / MILLI);

            for (w_id = aptr->starting_warehouse; w_id <=
aptr->num_warehouses; w_id++)
            {
                        d_w_id = w_id;

                        for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE;
d_id++)
                        {
                                    MakeAlphaString(6,10,D_NAME_LEN,
d_name);

                                    MakeAddress(d_street_1, d_street_2, d_city,
d_state, d_zip);

                                    d_tax = ((float)
RandomNumber(0L,2000L))/10000.00;

                                    rc = bcp_sendrow(w_hdbc1);
                                    if (rc != SUCCEED)
                                                HandleErrorDBC(w_hdbc1);

                                    district_rows_loaded++;
                                    CheckForCommit(w_hdbc1, w_hstmt1,
district_rows_loaded, "district", &time_start);
                        }
    }

            rcint = bcp_done(w_hdbc1);
            if (rcint < 0)
                        HandleErrorDBC(w_hdbc1);

            printf("Finished loading district table.\n");

            // if build index after load...
            if ((aptr->build_index == 1) && (aptr->index_order == 0))
                        BuildIndex("idxdiscl");

    return;
}


//=============================================================
================
//
// Function   : Stock
//
```

```
//==============================================================
================

void Stock()
{
   long     s_i_id;
   short    s_w_id;
   short    s_quantity;
   char     s_dist_01[S_DIST_LEN+1];
   char     s_dist_02[S_DIST_LEN+1];
   char     s_dist_03[S_DIST_LEN+1];
   char     s_dist_04[S_DIST_LEN+1];
   char     s_dist_05[S_DIST_LEN+1];
   char     s_dist_06[S_DIST_LEN+1];
   char     s_dist_07[S_DIST_LEN+1];
   char     s_dist_08[S_DIST_LEN+1];
   char     s_dist_09[S_DIST_LEN+1];
   char     s_dist_10[S_DIST_LEN+1];
   long     s_ytd;
   short    s_order_cnt;
   short    s_remote_cnt;
   char     s_data[S_DATA_LEN+1];
   short    len;
            char        name[20];
            long        time_start;
            RETCODE             rc;
            DBINT       rcint;
            char        bcphint[128];

            // Seed with unique number
            seed(3);

            // if build index before load...
            if ((aptr->build_index == 1) && (aptr->index_order == 1))
                        BuildIndex("idxstkcl");

            sprintf(name, "%s..%s", aptr->database, "stock");

            rc = bcp_init(w_hdbc1, name, NULL, "logs\\stock.err", DB_IN);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            if ((aptr->build_index == 1) && (aptr->index_order == 1))
            {
                        sprintf(bcphint, "tablock, order (s_i_id, s_w_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 100000));
                        rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
                        if (rc != SUCCEED)
                                    HandleErrorDBC(w_hdbc1);
            }

            rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 2);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) &s_quantity, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01, 0, S_DIST_LEN,
NULL, 0, 0, 4);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02, 0, S_DIST_LEN,
NULL, 0, 0, 5);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03, 0, S_DIST_LEN,
NULL, 0, 0, 6);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04, 0, S_DIST_LEN,
NULL, 0, 0, 7);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05, 0, S_DIST_LEN,
NULL, 0, 0, 8);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06, 0, S_DIST_LEN,
NULL, 0, 0, 9);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07, 0, S_DIST_LEN,
NULL, 0, 0, 10);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0, S_DIST_LEN,
NULL, 0, 0, 11);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09, 0, S_DIST_LEN,
NULL, 0, 0, 12);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0, S_DIST_LEN,
NULL, 0, 0, 13);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 14);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) &s_order_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 15);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) &s_remote_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 16);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0, S_DATA_LEN, NULL,
0, 0, 17);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            s_ytd = s_order_cnt = s_remote_cnt = 0;

            time_start = (TimeNow() / MILLI);

            printf("...Loading stock table\n");
```

```
        for (s_i_id=1; s_i_id <= max_items; s_i_id++)
        {

                for (s_w_id = (short)aptr->starting_warehouse; s_w_id <=
aptr->num_warehouses; s_w_id++)
                {

                        s_quantity =
(short)RandomNumber(10L,100L);
                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_01);
                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_02);
                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_03);
                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_04);
                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_05);
                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_06);
                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_07);
                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_08);
                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_09);
                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_10);

                        len = MakeOriginalAlphaString(26,50,
S_DATA_LEN, s_data,10);

                        rc = bcp_sendrow(w_hdbc1);
                        if (rc != SUCCEED)
                                HandleErrorDBC(w_hdbc1);

                        stock_rows_loaded++;
                        CheckForCommit(w_hdbc1, w_hstmt1,
stock_rows_loaded, "stock", &time_start);

                }
        }

        rcint = bcp_done(w_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(w_hdbc1);

        printf("Finished loading stock table.\n");

        SQLFreeStmt(w_hstmt1, SQL_DROP);
        SQLDisconnect(w_hdbc1);
        SQLFreeConnect(w_hdbc1);

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxstkcl");

        return;
}


//=============================================================
================
//
// Function   : LoadCustomer
//
//=============================================================
================

void LoadCustomer()
{
        LOADER_TIME_STRUCT      customer_time_start;
        LOADER_TIME_STRUCT      history_time_start;
        short                   w_id;
  short         d_id;
        DWORD
dwThreadID[MAX_CUSTOMER_THREADS];
        HANDLE
hThread[MAX_CUSTOMER_THREADS];
        char                    name[20];
        RETCODE
rc;
        DBINT                                           rcint;
        char
bcphint[128];
        char                                            cmd[256];
        // SQLRETURN                                    rc_1;
        // SQLSMALLINT                                  recnum,
MsgLen;
        // SQLCHAR
SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
        // SQLINTEGER                                   NativeError;

        // Seed with unique number
        seed(5);

        printf("Loading customer and history tables...\n");

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
                BuildIndex("idxcuscl");

        // Initialize bulk copy
        sprintf(name, "%s..%s", aptr->database, "customer");

        rc = bcp_init(c_hdbc1, name, NULL, "logs\\customer.err", DB_IN);

        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (c_w_id, c_d_id, c_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
                rc = bcp_control(c_hdbc1, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);
        }

        sprintf(name, "%s..%s", aptr->database, "history");

        rc = bcp_init(c_hdbc2, name, NULL, "logs\\history.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

        sprintf(bcphint, "tablock");
        rc = bcp_control(c_hdbc2, BCPHINTS, (void*) bcphint);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

        customer_rows_loaded    = 0;
        history_rows_loaded     = 0;

        CustomerBufInit();

        customer_time_start.time_start = (TimeNow() / MILLI);
        history_time_start.time_start = (TimeNow() / MILLI);

        for (w_id = (short)aptr->starting_warehouse; w_id <=
aptr->num_warehouses; w_id++)
        {
```

```
                for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE;
d_id++)
                {
                        CustomerBufLoad(d_id, w_id);

                        // Start parallel loading threads here...

                        // Start customer table thread

                        printf("...Loading customer table for: d_id =
%d, w_id = %d\n", d_id, w_id);

                        hThread[0] = CreateThread(NULL,
        0,
        (LPTHREAD_START_ROUTINE) LoadCustomerTable,
        &customer_time_start,
        0,
        &dwThreadID[0]);

                        if (hThread[0] == NULL)
                        {
                                printf("Error, failed in creating
creating thread = 0.\n");
                                exit(-1);
                        }

                        // Start History table thread

                        printf("...Loading history table for: d_id = %d,
w_id = %d\n", d_id, w_id);

                        hThread[1] = CreateThread(NULL,
        0,
        (LPTHREAD_START_ROUTINE) LoadHistoryTable,
        &history_time_start,
        0,
        &dwThreadID[1]);

                        if (hThread[1] == NULL)
                        {
                                printf("Error, failed in creating
creating thread = 1.\n");
                                exit(-1);
                        }

                        WaitForSingleObject( hThread[0], INFINITE
);
                        WaitForSingleObject( hThread[1], INFINITE
);

                        if (CloseHandle(hThread[0]) == FALSE)
                        {
                                printf("Error, failed in closing
customer thread handle with errno: %d\n", GetLastError());
                        }

                        if (CloseHandle(hThread[1]) == FALSE)
                        {
                                printf("Error, failed in closing
history thread handle with errno: %d\n", GetLastError());
                        }


                }

        }

        // flush the bulk connection
        rcint = bcp_done(c_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(c_hdbc1);

        rcint = bcp_done(c_hdbc2);
        if (rcint < 0)
                HandleErrorDBC(c_hdbc2);

        printf("Finished loading customer table.\n");

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxcuscl");

        // build non-clustered index
        if (aptr->build_index == 1)
                BuildIndex("idxcusnc");

        // Output the NURAND used for the loader into C_FIRST for C_ID =
1,
        // C_W_ID = 1, and C_D_ID = 1
        sprintf(cmd, "isql -S%s -U%s -P%s -d%s -e -Q\"update customer set
c_first = 'C_LOAD = %d' where c_id = 1 and c_w_id = 1 and c_d_id = 1\" >
logs\\nurand_load.log",
                                aptr->server,
                                aptr->user,
                                aptr->password,
                                aptr->database,
                                LOADER_NURAND_C);

        system(cmd);

        SQLFreeStmt(c_hstmt1, SQL_DROP);
        SQLDisconnect(c_hdbc1);
        SQLFreeConnect(c_hdbc1);

        SQLFreeStmt(c_hstmt2, SQL_DROP);
        SQLDisconnect(c_hdbc2);
        SQLFreeConnect(c_hdbc2);


   return;

}


//=================================================
================
//
// Function   : CustomerBufInit
//
//=================================================
================

void CustomerBufInit()
{
        int     i;

        for (i=0;i<customers_per_district;i++)
        {
                customer_buf[i].c_id = 0;
                customer_buf[i].c_d_id = 0;
                customer_buf[i].c_w_id = 0;
```

```
                strcpy(customer_buf[i].c_first,"");
                strcpy(customer_buf[i].c_middle,"");
                strcpy(customer_buf[i].c_last,"");
                strcpy(customer_buf[i].c_street_1,"");
                strcpy(customer_buf[i].c_street_2,"");
                strcpy(customer_buf[i].c_city,"");
                strcpy(customer_buf[i].c_state,"");
                strcpy(customer_buf[i].c_zip,"");
                strcpy(customer_buf[i].c_phone,"");
                strcpy(customer_buf[i].c_credit,"");

                customer_buf[i].c_credit_lim = 0;
                customer_buf[i].c_discount = (float) 0;

                //  fix to avoid ODBC float to numeric conversion problem.
                //              customer_buf[i].c_balance = 0;
                strcpy(customer_buf[i].c_balance,"");

                customer_buf[i].c_ytd_payment = 0;
                customer_buf[i].c_payment_cnt = 0;
                customer_buf[i].c_delivery_cnt = 0;

                strcpy(customer_buf[i].c_data,"");

                customer_buf[i].h_amount = 0;

                strcpy(customer_buf[i].h_data,"");

        }


}


//=========================================================
================
//
// Function   : CustomerBufLoad
//
// Fills shared buffer for HISTORY and CUSTOMER
//=========================================================
================

void CustomerBufLoad(int d_id, int w_id)
{
        long                                            i;
  CUSTOMER_SORT_STRUCT    c[CUSTOMERS_PER_DISTRICT];


        for (i=0;i<customers_per_district;i++)
        {
                if (i < 1000)
                        LastName(i, c[i].c_last);
                else

LastName(NURand(255,0,999,LOADER_NURAND_C), c[i].c_last);

                MakeAlphaString(8,16,FIRST_NAME_LEN, c[i].c_first);

                c[i].c_id = i+1;

        }

        printf("...Loading customer buffer for: d_id = %d, w_id = %d\n",
                        d_id, w_id);

        for (i=0;i<customers_per_district;i++)
        {

                customer_buf[i].c_d_id = d_id;
                customer_buf[i].c_w_id = w_id;
                customer_buf[i].h_amount = 10.0;

                customer_buf[i].c_ytd_payment  = 10.0;

                customer_buf[i].c_payment_cnt  = 1;
                customer_buf[i].c_delivery_cnt = 0;

                // Generate CUSTOMER and HISTORY data

                customer_buf[i].c_id = c[i].c_id;

                strcpy(customer_buf[i].c_first, c[i].c_first);
                strcpy(customer_buf[i].c_last, c[i].c_last);

                customer_buf[i].c_middle[0] = 'O';
                customer_buf[i].c_middle[1] = 'E';

                MakeAddress(customer_buf[i].c_street_1,
                                customer_buf[i].c_street_2,
                                customer_buf[i].c_city,
                                customer_buf[i].c_state,
                                customer_buf[i].c_zip);

                MakeNumberString(16, 16, PHONE_LEN,
customer_buf[i].c_phone);

                if (RandomNumber(1L, 100L) > 10)
                        customer_buf[i].c_credit[0] = 'G';
                else
                        customer_buf[i].c_credit[0] = 'B';
                customer_buf[i].c_credit[1] = 'C';

                customer_buf[i].c_credit_lim = 50000.0;
                customer_buf[i].c_discount = ((float) RandomNumber(0L,
5000L)) / 10000.0;

                //  fix to avoid ODBC float to numeric conversion problem.

                //  customer_buf[i].c_balance = -10.0;
                strcpy(customer_buf[i].c_balance,"-10.0");

                MakeAlphaString(300, 500, C_DATA_LEN,
customer_buf[i].c_data);

                // Generate HISTORY data
                MakeAlphaString(12, 24, H_DATA_LEN,
customer_buf[i].h_data);

        }
}

//=========================================================
================
//
// Function   : LoadCustomerTable
//
//=========================================================
================

void LoadCustomerTable(LOADER_TIME_STRUCT *customer_time_start)
{
        int     i;
  long      c_id;
  short     c_d_id;
  short     c_w_id;
  char      c_first[FIRST_NAME_LEN+1];
  char      c_middle[MIDDLE_NAME_LEN+1];
  char      c_last[LAST_NAME_LEN+1];
  char      c_street_1[ADDRESS_LEN+1];
  char      c_street_2[ADDRESS_LEN+1];
```

```
    char        c_city[ADDRESS_LEN+1];
    char        c_state[STATE_LEN+1];
    char        c_zip[ZIP_LEN+1];
    char        c_phone[PHONE_LEN+1];
    char        c_credit[CREDIT_LEN+1];
    double      c_credit_lim;
    double      c_discount;

                //  fix to avoid ODBC float to numeric conversion problem.
                //  double              c_balance;
                char                    c_balance[6];

    double      c_ytd_payment;
    short       c_payment_cnt;
    short       c_delivery_cnt;
    char        c_data[C_DATA_LEN+1];
                char                    c_since[C_SINCE_LEN+1];
                RETCODE                         rc;

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 1);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 2);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

                rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0, FIRST_NAME_LEN, NULL, 0,
0, 4);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0,
MIDDLE_NAME_LEN,NULL, 0, 0, 5);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN, NULL, 0, 0,
6);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN, NULL, 0,
0, 7);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN, NULL,0,0,
8);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL, 0, 0, 9);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0, 0, 10);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0, 11);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL, 0, 0,
12);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

                rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0, C_SINCE_LEN,
NULL, 0, SQLCHARACTER, 13);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL, 0, 0,
14);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 15);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 16);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);


                //  fix to avoid ODBC float to numeric conversion problem.

    //  rc = bcp_bind(c_hdbc1, (BYTE *) &c_balance, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 17);
                //  if (rc != SUCCEED)
                //          HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0,
SQLCHARACTER, 17);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);


    rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 18);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 19);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt,0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 20);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0, 500, NULL, 0, 0, 21);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    for (i = 0; i < customers_per_district; i++)
    {
                        c_id = customer_buf[i].c_id;
                        c_d_id = customer_buf[i].c_d_id;
                        c_w_id = customer_buf[i].c_w_id;

                        strcpy(c_first, customer_buf[i].c_first);
                        strcpy(c_middle, customer_buf[i].c_middle);
                        strcpy(c_last, customer_buf[i].c_last);
                        strcpy(c_street_1, customer_buf[i].c_street_1);
                        strcpy(c_street_2, customer_buf[i].c_street_2);
                        strcpy(c_city, customer_buf[i].c_city);
                        strcpy(c_state, customer_buf[i].c_state);
                        strcpy(c_zip, customer_buf[i].c_zip);
```

```
                        strcpy(c_phone, customer_buf[i].c_phone);
                        strcpy(c_credit, customer_buf[i].c_credit);

                        FormatDate(&c_since);


                        c_credit_lim = customer_buf[i].c_credit_lim;
                        c_discount = customer_buf[i].c_discount;

                        //  fix to avoid ODBC float to numeric conversion problem.

                        //  c_balance = customer_buf[i].c_balance;
                        strcpy(c_balance, customer_buf[i].c_balance);

                        c_ytd_payment = customer_buf[i].c_ytd_payment;
                        c_payment_cnt = customer_buf[i].c_payment_cnt;
                        c_delivery_cnt = customer_buf[i].c_delivery_cnt;

                        strcpy(c_data, customer_buf[i].c_data);

                        // Send data to server
                        rc = bcp_sendrow(c_hdbc1);
                        if (rc != SUCCEED)
                                HandleErrorDBC(c_hdbc1);

                        customer_rows_loaded++;
                        CheckForCommit(c_hdbc1, c_hstmt1,
customer_rows_loaded, "customer", &customer_time_start->time_start);
    }

}


//=====================================================
================
//
// Function   : LoadHistoryTable
//
//=====================================================
================

void LoadHistoryTable(LOADER_TIME_STRUCT *history_time_start)
{
        int     i;
    long    c_id;
    short   c_d_id;
    short   c_w_id;
        double  h_amount;
    char    h_data[H_DATA_LEN+1];
        char                    h_date[H_DATE_LEN+1];
        RETCODE                         rc;

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

        rc = bcp_bind(c_hdbc2, (BYTE *) &h_date, 0, H_DATE_LEN,
NULL, 0, SQLCHARACTER, 6);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 7);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0, H_DATA_LEN, NULL, 0, 0, 8);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

    for (i = 0; i < customers_per_district; i++)
    {
                        c_id = customer_buf[i].c_id;
                        c_d_id = customer_buf[i].c_d_id;
                        c_w_id = customer_buf[i].c_w_id;
                        h_amount = customer_buf[i].h_amount;
                        strcpy(h_data, customer_buf[i].h_data);

                        FormatDate(&h_date);

                        // send to server
                        rc = bcp_sendrow(c_hdbc2);
                        if (rc != SUCCEED)
                                HandleErrorDBC(c_hdbc2);

                        history_rows_loaded++;
                        CheckForCommit(c_hdbc2, c_hstmt2,
history_rows_loaded, "history", &history_time_start->time_start);
    }

}


//=====================================================
======-================
//
// Function   : LoadOrders
//
//=====================================================
=====================

void LoadOrders()
{
        LOADER_TIME_STRUCT      orders_time_start;
        LOADER_TIME_STRUCT      new_order_time_start;
        LOADER_TIME_STRUCT      order_line_time_start;
        short                           w_id;
    short               d_id;
        DWORD
dwThreadID[MAX_ORDER_THREADS];
        HANDLE
hThread[MAX_ORDER_THREADS];
        char                            name[20];
        RETCODE
rc;
        char
bcphint[128];

        // seed with unique number
        seed(6);
```

```
	printf("Loading orders...\n");

	// if build index before load...
	if ((aptr->build_index == 1) && (aptr->index_order == 1))
	{
			BuildIndex("idxordcl");
			BuildIndex("idxnodcl");
			BuildIndex("idxodlcl");
	}

	// initialize bulk copy
	sprintf(name, "%s..%s", aptr->database, "orders");

	rc = bcp_init(o_hdbc1, name, NULL, "logs\\orders.err", DB_IN);
	if (rc != SUCCEED)
		HandleErrorDBC(o_hdbc1);

	if ((aptr->build_index == 1) && (aptr->index_order == 1))
	{
		sprintf(bcphint, "tablock, order (o_w_id, o_d_id, o_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
		rc = bcp_control(o_hdbc1, BCPHINTS, (void*) bcphint);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc1);
	}

	sprintf(name, "%s..%s", aptr->database, "new_order");

	rc = bcp_init(o_hdbc2, name, NULL, "logs\\neword.err", DB_IN);
	if (rc != SUCCEED)
		HandleErrorDBC(o_hdbc2);

	if ((aptr->build_index == 1) && (aptr->index_order == 1))
	{
		sprintf(bcphint, "tablock, order (no_w_id, no_d_id,
no_o_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 9000));
		rc = bcp_control(o_hdbc2, BCPHINTS, (void*) bcphint);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc2);
	}

	sprintf(name, "%s..%s", aptr->database, "order_line");

	rc = bcp_init(o_hdbc3, name, NULL, "logs\\ordline.err", DB_IN);
	if (rc != SUCCEED)
		HandleErrorDBC(o_hdbc3);

  if ((aptr->build_index == 1) && (aptr->index_order == 1))
	{
		sprintf(bcphint, "tablock, order (ol_w_id, ol_d_id, ol_o_id,
ol_number), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 300000));
		rc = bcp_control(o_hdbc3, BCPHINTS, (void*) bcphint);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc3);
	}

	orders_rows_loaded      = 0;
	new_order_rows_loaded   = 0;
	order_line_rows_loaded  = 0;

	OrdersBufInit();

	orders_time_start.time_start = (TimeNow() / MILLI);
	new_order_time_start.time_start = (TimeNow() / MILLI);
	order_line_time_start.time_start = (TimeNow() / MILLI);

	for (w_id = (short)aptr->starting_warehouse; w_id <=
aptr->num_warehouses; w_id++)
	{
		for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE;
d_id++)
		{
			OrdersBufLoad(d_id, w_id);

			// start parallel loading threads here...

			// start Orders table thread

			printf("...Loading Order Table for: d_id = %d,
w_id = %d\n", d_id, w_id);

			hThread[0] = CreateThread(NULL,
	0,
	(LPTHREAD_START_ROUTINE) LoadOrdersTable,
	&orders_time_start,
	0,
	&dwThreadID[0]);

			if (hThread[0] == NULL)
			{
				printf("Error, failed in creating
creating thread = 0.\n");
				exit(-1);
			}

			// start NewOrder table thread

			printf("...Loading New-Order Table for: d_id =
%d, w_id = %d\n", d_id, w_id);

			hThread[1] = CreateThread(NULL,
	0,
	(LPTHREAD_START_ROUTINE) LoadNewOrderTable,
	&new_order_time_start,
	0,
	&dwThreadID[1]);

			if (hThread[1] == NULL)
			{
				printf("Error, failed in creating
creating thread = 1.\n");
				exit(-1);
			}

			// start Order-Line table thread

			printf("...Loading Order-Line Table for: d_id =
%d, w_id = %d\n", d_id, w_id);

			hThread[2] = CreateThread(NULL,
	0,
	(LPTHREAD_START_ROUTINE) LoadOrderLineTable,
	&order_line_time_start,
	0,
	&dwThreadID[2]);
```

```
                                if (hThread[2] == NULL)
                                {
                                        printf("Error, failed in creating
creating thread = 2.\n");
                                        exit(-1);
                                }

                                WaitForSingleObject( hThread[0], INFINITE
);
                                WaitForSingleObject( hThread[1], INFINITE
);
                                WaitForSingleObject( hThread[2], INFINITE
);

                                if (CloseHandle(hThread[0]) == FALSE)
                                {
                                        printf("Error, failed in closing
Orders thread handle with errno: %d\n", GetLastError());
                                }

                                if (CloseHandle(hThread[1]) == FALSE)
                                {
                                        printf("Error, failed in closing
NewOrder thread handle with errno: %d\n", GetLastError());
                                }

                                if (CloseHandle(hThread[2]) == FALSE)
                                {
                                        printf("Error, failed in closing
OrderLine thread handle with errno: %d\n", GetLastError());
                                }

                        }
                }

                printf("Finished loading orders.\n");


   return;
}


//==============================================================
================
//
// Function   : OrdersBufInit
//
// Clears shared buffer for ORDERS, NEWORDER, and ORDERLINE
//
//==============================================================
================

void OrdersBufInit()
{
        int     i;
        int                     j;

        for (i=0;i<orders_per_district;i++)
        {
                orders_buf[i].o_id = 0;
                orders_buf[i].o_d_id = 0;
                orders_buf[i].o_w_id = 0;
                orders_buf[i].o_c_id = 0;
                orders_buf[i].o_carrier_id = 0;
                orders_buf[i].o_ol_cnt = 0;
                orders_buf[i].o_all_local = 0;

                for (j=0;j<=14;j++)
                {
                        orders_buf[i].o_ol[j].ol = 0;
                        orders_buf[i].o_ol[j].ol_i_id = 0;

                        orders_buf[i].o_ol[j].ol_supply_w_id = 0;
                        orders_buf[i].o_ol[j].ol_quantity = 0;
                        orders_buf[i].o_ol[j].ol_amount = 0;
                        strcpy(orders_buf[i].o_ol[j].ol_dist_info,"");
                }
        }


}


//==============================================================
================
//
// Function   : OrdersBufLoad
//
// Fills shared buffer for ORDERS, NEWORDER, and ORDERLINE
//
//==============================================================
================

void OrdersBufLoad(int d_id, int w_id)
{

        int     cust[ORDERS_PER_DISTRICT+1];
        long    o_id;
        short   ol;

        printf("...Loading Order Buffer for: d_id = %d, w_id = %d\n",
                        d_id, w_id);

        GetPermutation(cust, orders_per_district);

        for (o_id=0;o_id<orders_per_district;o_id++)
        {

                // Generate ORDER and NEW-ORDER data

                orders_buf[o_id].o_d_id = d_id;
                orders_buf[o_id].o_w_id = w_id;
                orders_buf[o_id].o_id = o_id+1;
                orders_buf[o_id].o_c_id = cust[o_id+1];
                orders_buf[o_id].o_ol_cnt = (short)RandomNumber(5L,
15L);

                if (o_id < first_new_order)
                {
                        orders_buf[o_id].o_carrier_id =
(short)RandomNumber(1L, 10L);
                        orders_buf[o_id].o_all_local  = 1;
                }
                else
                {
                        orders_buf[o_id].o_carrier_id = 0;
                        orders_buf[o_id].o_all_local  = 1;
                }

                for (ol=0; ol<orders_buf[o_id].o_ol_cnt; ol++)
                {

                        orders_buf[o_id].o_ol[ol].ol = ol+1;
                        orders_buf[o_id].o_ol[ol].ol_i_id =
RandomNumber(1L, max_items);
                        orders_buf[o_id].o_ol[ol].ol_supply_w_id =
w_id;
                        orders_buf[o_id].o_ol[ol].ol_quantity = 5;
                        MakeAlphaString(24, 24,
OL_DIST_INFO_LEN, &orders_buf[o_id].o_ol[ol].ol_dist_info);
```

```
                                        // Generate ORDER-LINE data
                                        if (o_id < first_new_order)
                                        {


orders_buf[o_id].o_ol[ol].ol_amount = 0;
                                                        // Added to insure ol_delivery_d set
properly during load


FormatDate(&orders_buf[o_id].o_ol[ol].ol_delivery_d);

                                        }
                                        else
                                        {

orders_buf[o_id].o_ol[ol].ol_amount = RandomNumber(1,999999)/100.0;

                                                        // Added to insure ol_delivery_d set
properly during load

                                                        // odbc datetime format

strcpy(orders_buf[o_id].o_ol[ol].ol_delivery_d,"1899-12-31 00:00:00.000");

                                        }
                                }
                        }
}


//=======================================================
================
//
// Function   : LoadOrdersTable
//
//=======================================================
================

void LoadOrdersTable(LOADER_TIME_STRUCT *orders_time_start)
{
                int             i;
   long         o_id;
                short           o_d_id;
                short           o_w_id;
   long         o_c_id;
   short        o_carrier_id;
   short        o_ol_cnt;
   short        o_all_local;
                char                    o_entry_d[O_ENTRY_D_LEN+1];
                RETCODE                         rc;
                DBINT                   rcint;

                // bind ORDER data
   rc = bcp_bind(o_hdbc1, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 1);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);

   rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 2);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);

   rc = bcp_bind(o_hdbc1, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 3);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);

   rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT4, 4);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);

                rc = bcp_bind(o_hdbc1, (BYTE *) &o_entry_d, 0,
O_ENTRY_D_LEN, NULL, 0, SQLCHARACTER, 5);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);

   rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 6);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);

   rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 7);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);

   rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 8);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);

   for (i = 0; i < orders_per_district; i++)
   {
                        o_id        = orders_buf[i].o_id;
                        o_d_id      = orders_buf[i].o_d_id;
                        o_w_id      = orders_buf[i].o_w_id;
                        o_c_id      = orders_buf[i].o_c_id;
                        o_carrier_id = orders_buf[i].o_carrier_id;
                        o_ol_cnt    = orders_buf[i].o_ol_cnt;
                        o_all_local  = orders_buf[i].o_all_local;

                        FormatDate(&o_entry_d);

                        // send data to server
                        rc = bcp_sendrow(o_hdbc1);
                        if (rc != SUCCEED)
                                HandleErrorDBC(o_hdbc1);

                        orders_rows_loaded++;
                        CheckForCommit(o_hdbc1, o_hstmt1,
orders_rows_loaded, "orders", &orders_time_start->time_start);
   }

                // rcint = bcp_batch(o_hdbc1);
                // if (rcint < 0)
                //              HandleErrorDBC(o_hdbc1);

                if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
                {
                        rcint = bcp_done(o_hdbc1);
                        if (rcint < 0)
                                HandleErrorDBC(o_hdbc1);

                        SQLFreeStmt(o_hstmt1, SQL_DROP);
                        SQLDisconnect(o_hdbc1);
                        SQLFreeConnect(o_hdbc1);

                        // if build index after load...
                        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                                BuildIndex("idxordcl");

                        // build non-clustered index
                        if (aptr->build_index == 1)
                                BuildIndex("idxordnc");
                }

}
```

```
//==============================================================
//
// Function   : LoadNewOrderTable
//
//==============================================================

void LoadNewOrderTable(LOADER_TIME_STRUCT *new_order_time_start)
{
   int        i;
   long       o_id;
   short      o_d_id;
   short      o_w_id;
              RETCODE                   rc;
              DBINT              rcint;

              // Bind NEW-ORDER data

   rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 1);
              if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc2);

   rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 2);
              if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc2);

   rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 3);
              if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc2);

              for (i = first_new_order; i < last_new_order; i++)
              {
                        o_id    = orders_buf[i].o_id;
                        o_d_id  = orders_buf[i].o_d_id;
                        o_w_id  = orders_buf[i].o_w_id;

                        rc = bcp_sendrow(o_hdbc2);
                        if (rc != SUCCEED)
                                  HandleErrorDBC(o_hdbc2);

                        new_order_rows_loaded++;
                        CheckForCommit(o_hdbc2, o_hstmt2,
new_order_rows_loaded, "new_order", &new_order_time_start->time_start);
              }

              // rcint = bcp_batch(o_hdbc2);
              // if (rcint < 0)
              //           HandleErrorDBC(o_hdbc2);

              if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
              {
                        rcint = bcp_done(o_hdbc2);
                        if (rcint < 0)
                                  HandleErrorDBC(o_hdbc2);

                        SQLFreeStmt(o_hstmt2, SQL_DROP);
                        SQLDisconnect(o_hdbc2);
                        SQLFreeConnect(o_hdbc2);

                        // if build index after load...
                        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                                  BuildIndex("idxnodcl");

              }

}
```

```
//==============================================================
//
// Function   : LoadOrderLineTable
//
//==============================================================

void LoadOrderLineTable(LOADER_TIME_STRUCT *order_line_time_start)
{
              int        i,j;
   long       o_id;
              short      o_d_id;
              short      o_w_id;
   long       ol;
              long       ol_i_id;
   short      ol_supply_w_id;
   short      ol_quantity;
   double     ol_amount;
   char       ol_dist_info[DIST_INFO_LEN+1];
              char                 ol_delivery_d[OL_DELIVERY_D_LEN+1];
              RETCODE                   rc;
              DBINT              rcint;

              // bind ORDER-LINE data
   rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 1);
              if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

   rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 2);
              if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

   rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 3);
              if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

   rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 4);
              if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

   rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT4, 5);
              if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

   rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 6);
              if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

              rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0,
OL_DELIVERY_D_LEN, NULL, 0, SQLCHARACTER, 7);
              if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

   rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 8);
              if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

   rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 9);
              if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
```

```
   rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0, DIST_INFO_LEN, NULL,
0, 0, 10);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

            for (i = 0; i < orders_per_district; i++)
   {
                        o_id    = orders_buf[i].o_id;
                        o_d_id  = orders_buf[i].o_d_id;
                        o_w_id  = orders_buf[i].o_w_id;

                        for (j=0; j < orders_buf[i].o_ol_cnt; j++)
                        {
                                    ol          = orders_buf[i].o_ol[j].ol;
                                    ol_i_id     = orders_buf[i].o_ol[j].ol_i_id;
                                    ol_supply_w_id =
orders_buf[i].o_ol[j].ol_supply_w_id;
                                    ol_quantity    =
orders_buf[i].o_ol[j].ol_quantity;
                                    ol_amount      =
orders_buf[i].o_ol[j].ol_amount;

strcpy(ol_delivery_d,orders_buf[i].o_ol[j].ol_delivery_d);


strcpy(ol_dist_info,orders_buf[i].o_ol[j].ol_dist_info);

                                    rc = bcp_sendrow(o_hdbc3);
                                    if (rc != SUCCEED)
                                                HandleErrorDBC(o_hdbc3);

                                    order_line_rows_loaded++;
                                    CheckForCommit(o_hdbc3, o_hstmt3,
order_line_rows_loaded, "order_line", &order_line_time_start->time_start);
                        }

            }

            // rcint = bcp_batch(o_hdbc3);
            // if (rcint < 0)
            //          HandleErrorDBC(o_hdbc3);

            if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
            {
                        rcint = bcp_done(o_hdbc3);
                        if (rcint < 0)
                                    HandleErrorDBC(o_hdbc3);

                        SQLFreeStmt(o_hstmt3, SQL_DROP);
                        SQLDisconnect(o_hdbc3);
                        SQLFreeConnect(o_hdbc3);

                        // if build index after load...
                        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                                    BuildIndex("idxodlcl");

            }

}


//=================================================================
================
//
// Function   : GetPermutation
//
//=================================================================
================

void GetPermutation(int perm[], int n)
{
            int i, r, t;

            for (i=1;i<=n;i++)
                        perm[i] = i;

            for (i=1;i<=n;i++)
            {
                        r = RandomNumber(i,n);
                        t = perm[i];
                        perm[i] = perm[r];
                        perm[r] = t;
            }

}


//=================================================================
================
//
// Function   : CheckForCommit
//
//=================================================================
================

void CheckForCommit(HDBC hdbc,
                                                HSTMT hstmt,
                                    int rows_loaded,
                                                char *table_name,
                                    long *time_start)
{

   long     time_end, time_diff;
            // DBINT   rcint;

   if ( !(rows_loaded % aptr->batch) )
   {

                        // rcint = bcp_batch(hdbc);
                        // if (rcint < 0)
                        //          HandleErrorDBC(hdbc);

                        time_end = (TimeNow() / MILLI);
                        time_diff = time_end - *time_start;

                        printf("-> Loaded %ld rows into %s in %ld sec - Total =
%d (%.2f rps)\n",
                                    aptr->batch,
                                    table_name,
                                                time_diff,
                                    rows_loaded,
                                                (float) aptr->batch / (time_diff ?
time_diff : 1L));

                        *time_start = time_end;
   }

   return;
}


//=================================================================
================
//
// Function   : OpenConnections
//
//=================================================================
================

void OpenConnections()
{
```

```
	RETCODE		rc;

	char				szDriverString[300];
	char				szDriverStringOut[1024];
	SQLSMALLINT			cbDriverStringOut;


	SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE,
&henv );

	SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION,
(void*)SQL_OV_ODBC3, 0 );

	SQLAllocHandle(SQL_HANDLE_DBC, henv , &i_hdbc1);
	SQLAllocHandle(SQL_HANDLE_DBC, henv , &w_hdbc1);
	SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc1);
	SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc2);
	SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc1);
	SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc2);
	SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc3);

	SQLSetConnectAttr(i_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
	SQLSetConnectAttr(w_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
	SQLSetConnectAttr(c_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
	SQLSetConnectAttr(c_hdbc2, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
	SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
	SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
	SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );

	// Open connections to SQL Server

	// Connection 1

	sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,
							aptr->user,

aptr->password,

aptr->database );

	rc = SQLSetConnectOption (i_hdbc1, SQL_PACKET_SIZE,
aptr->pack_size);
	if (rc != SUCCEED)
		HandleErrorDBC(i_hdbc1);

	rc = SQLDriverConnect ( i_hdbc1,
						NULL,

(SQLCHAR*)&szDriverString[0] ,
							SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );
	if (rc != SUCCEED)
		HandleErrorDBC(i_hdbc1);

	// Connection 2

	sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,
							aptr->user,

aptr->password,

aptr->database );

	rc = SQLSetConnectOption (w_hdbc1, SQL_PACKET_SIZE,
aptr->pack_size);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = SQLDriverConnect ( w_hdbc1,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	// Connection 3

	sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,
							aptr->user,

aptr->password,

aptr->database );

	rc = SQLSetConnectOption (c_hdbc1, SQL_PACKET_SIZE,
aptr->pack_size);
	if (rc != SUCCEED)
		HandleErrorDBC(c_hdbc1);

	rc = SQLDriverConnect ( c_hdbc1,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );
	if (rc != SUCCEED)
		HandleErrorDBC(c_hdbc1);

	// Connection 4
```

```
        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,
                                                        aptr->user,

aptr->password,

aptr->database );

        rc = SQLSetConnectOption (c_hdbc2, SQL_PACKET_SIZE,
aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

        rc = SQLDriverConnect ( c_hdbc2,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

        // Connection 5

        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,
                                                        aptr->user,

aptr->password,

aptr->database );

        rc = SQLSetConnectOption (o_hdbc1, SQL_PACKET_SIZE,
aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

        rc = SQLDriverConnect ( o_hdbc1,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

        // Connection 6

        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,
                                                        aptr->user,

aptr->password,

aptr->database );

        rc = SQLSetConnectOption (o_hdbc2, SQL_PACKET_SIZE,
aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

        rc = SQLDriverConnect ( o_hdbc2,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

        // Connection 7

        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,
                                                        aptr->user,

aptr->password,

aptr->database );

        rc = SQLSetConnectOption (o_hdbc3, SQL_PACKET_SIZE,
aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

        rc = SQLDriverConnect ( o_hdbc3,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

}
```

```
//=========================================================
================
//
// Function name: BuildIndex
//
//=========================================================
================

void BuildIndex(char *index_script)
{
        char    cmd[256];

        printf("Starting index creation:  %s\n",index_script);

        sprintf(cmd, "isql -S%s -U%s -P%s -e -i%s\\%s.sql > logs\\%s.log",
                                        aptr->server,
                                        aptr->user,
                                        aptr->password,
                                        aptr->index_script_path,
                                        index_script,
                                        index_script);

        system(cmd);

        printf("Finished index creation:  %s\n",index_script);
}


void HandleErrorDBC (SQLHDBC  hdbc1)
{

        SQLCHAR                 SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
        SQLINTEGER  NativeError;
        SQLSMALLINT i, MsgLen;
        SQLRETURN   rc2;
        char            timebuf[128];
        char            datebuf[128];
        FILE            *fp1;

        i = 1;
        while (( rc2 = SQLGetDiagRec(SQL_HANDLE_DBC , hdbc1, i,
SqlState , &NativeError,
                                        Msg, sizeof(Msg) , &MsgLen ))
!= SQL_NO_DATA )
        {

                sprintf( szLastError , "%s" ,  Msg );

                _strtime(timebuf);
                _strdate(datebuf);

                printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

                fp1 = fopen("logs\\tpccldr.err","w");
                if (fp1 == NULL)
                        printf("ERROR:  Unable to open errorlog
file.\n");
                else
                {
                        fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);
                        fclose(fp1);
                }

                i++;

        }
}

void HandleErrorSTMT (HSTMT  hstmt1)
{

        SQLCHAR                 SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
        SQLINTEGER  NativeError;
        SQLSMALLINT i, MsgLen;
        SQLRETURN   rc2;
        char            timebuf[128];
        char            datebuf[128];
        FILE            *fp1;

        i = 1;
        while (( rc2 = SQLGetDiagRec(SQL_HANDLE_STMT , hstmt1, i,
SqlState , &NativeError,
                                        Msg, sizeof(Msg) , &MsgLen ))
!= SQL_NO_DATA )
        {

                sprintf( szLastError , "%s" ,  Msg );

                _strtime(timebuf);
                _strdate(datebuf);

                printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

                fp1 = fopen("logs\\tpccldr.err","w");
                if (fp1 == NULL)
                        printf("ERROR:  Unable to open errorlog
file.\n");
                else
                {
                        fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);
                        fclose(fp1);
                }

                i++;

        }
}


void FormatDate ( char* szTimeCOutput )
{

        struct tm when;
        time_t now;

        time( &now );
        when  = *localtime( &now );

        mktime( &when );

        // odbc datetime format
        strftime( szTimeCOutput , 30 , "%Y-%m-%d %H:%M:%S.000",
&when );

        return;

}
//=========================================================
================
//
// Function   : CheckDataBase
//
//=========================================================
================

void CheckDataBase()
{
```

```
	RETCODE	rc;

	char			szDriverString[300];
	char			szDriverStringOut[1024];
	char			TablesBitMap[9] =
{"000000000"};
	int				i, ExitFlag;

	SQLSMALLINT		cbDriverStringOut;
	SQLCHAR			TabName[10];
	SQLINTEGER		TabNameInd, TabCount,
TabCountInd;


	ExitFlag = 0;

	SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE,
&henv );

	SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION,
(void*)SQL_OV_ODBC3, 0 );

	SQLAllocHandle(SQL_HANDLE_DBC, henv , &v_hdbc);

	SQLSetConnectAttr(v_hdbc, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );

	// Open connection to SQL Server

	sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,
aptr->user,

aptr->password,

aptr->database );

	rc = SQLSetConnectAttr( v_hdbc, SQL_ATTR_PACKET_SIZE,
(SQLPOINTER)aptr->pack_size, SQL_IS_UINTEGER );
	if (rc != SQL_SUCCESS)
		HandleErrorDBC(v_hdbc);

	rc = SQLDriverConnect ( v_hdbc,
NULL,

(SQLCHAR*)&szDriverString[0] ,
SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );

	// if the rc is SQL_ERROR, the the TPCC database probably does not
exist
	if (rc == SQL_ERROR)
	{
		printf("The database TPCC does not appear to exist!\n");
		printf("\nCheck LOGS\\ directory for database creation
errors.\n");

		// cleanup database connections and handles
		SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
		SQLDisconnect(v_hdbc);
		SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);
		// since there is not a database, exit back to SETUP.CMD
		exit(1);
	}

	if ( SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt) !=
SQL_SUCCESS )
		HandleErrorDBC(v_hdbc);

	if ( SQLBindCol(v_hstmt, 1, SQL_C_ULONG, &TabCount, 0,
&TabCountInd) != SQL_SUCCESS )
		HandleErrorSTMT(v_hstmt);

	// count the number of user tables from sysobjects
	rc = SQLExecDirect(v_hstmt, "select count(*) from sysobjects where
xtype = \'U\'", SQL_NTS);
	if ((rc != SQL_SUCCESS) && (rc !=
SQL_SUCCESS_WITH_INFO))
		HandleErrorSTMT(v_hstmt);

	if ( SQLFetch(v_hstmt) != SQL_SUCCESS )
		HandleErrorSTMT(v_hstmt);

	// if the number of tables is less than 9, select all the user tables in
TPCC
	if (TabCount != 9)
	{
		SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);

		SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc ,
&v_hstmt);

		if ( SQLBindCol(v_hstmt, 1, SQL_C_CHAR, &TabName,
sizeof(TabName), &TabNameInd) != SQL_SUCCESS )
			HandleErrorSTMT(v_hstmt);

		// select the list of user tables into a result set
		rc = SQLExecDirect(v_hstmt, "select * from sysobjects
where xtype = \'U\'", SQL_NTS);
		if ((rc != SQL_SUCCESS) && (rc !=
SQL_SUCCESS_WITH_INFO))
			HandleErrorSTMT(v_hstmt);

		// go through the result set and set the bitmap for each
found table
		// set the bitmap to '1' if the table name is found

		while ((rc = SQLFetch(v_hstmt)) != SQL_NO_DATA)
		{
			switch( TabName[0] )
			{
			case 'w':
				TablesBitMap[0] = '1';
				break;
			case 'd':
				TablesBitMap[1] = '1';
				break;
			case 'c':
				TablesBitMap[2] = '1';
				break;
			case 'h':
				TablesBitMap[3] = '1';
				break;
			case 'n':
				TablesBitMap[4] = '1';
				break;
			case 'o':
				if (TabName[5] = 's')
					TablesBitMap[5] = '1';
				if (TabName[5] = '_')
					TablesBitMap[6] = '1';
```

```
                                        break;
                                case 'i':
                                        TablesBitMap[7] = '1';
                                        break;
                                case 's':
                                        TablesBitMap[8] = '1';
                                        break;
                                }
                        }

                        // a '0' ExitFlag means do NOT exit the loader early, a '1'
means exit the loader early
                        ExitFlag = 0;

                        // interate through the bitmap to display which table(s) is
actually missing
                        for (i = 0; i <= 8; i++)
                        {
                                switch(i)
                                {
                                case 0:
                                        if (TablesBitMap[i] == '0')
                                        {
                                                printf("The Warehouse
table is missing or damaged.\n");
                                                ExitFlag = 1;
                                        }
                                        break;
                                case 1:
                                        if (TablesBitMap[i] == '0')
                                        {
                                                printf("The District table
is missing or damaged.\n");
                                                ExitFlag = 1;
                                        }
                                        break;
                                case 2:
                                        if (TablesBitMap[i] == '0')
                                        {
                                                printf("The Customer
table is missing or damaged.\n");
                                                ExitFlag = 1;
                                        }
                                        break;
                                case 3:
                                        if (TablesBitMap[i] == '0')
                                        {
                                                printf("The History table
is missing or damaged.\n");
                                                ExitFlag = 1;
                                        }
                                        break;
                                case 4:
                                        if (TablesBitMap[i] == '0')
                                        {
                                                printf("The New_Order
table is missing or damaged.\n");
                                                ExitFlag = 1;
                                        }
                                        break;
                                case 5:
                                        if (TablesBitMap[i] == '0')
                                        {
                                                printf("The Orders table
is missing or damaged.\n");
                                                ExitFlag = 1;
                                        }
                                        break;
                                case 6:
                                        if (TablesBitMap[i] == '0')
                                        {
                                                printf("The Order_Line
table is missing or damaged.\n");
                                                ExitFlag = 1;
                                        }
                                        break;
                                case 7:
                                        if (TablesBitMap[i] == '0')
                                        {
                                                printf("The Item table is
missing or damaged.\n");
                                                ExitFlag = 1;
                                        }
                                        break;
                                case 8:
                                        if (TablesBitMap[i] == '0')
                                        {
                                                printf("The Stock table
is missing or damaged.\n");
                                                ExitFlag = 1;
                                        }
                                        break;
                                }
                        }

                        // if one or more tables are missing, display message and
exit the loader
                        if (ExitFlag = 1)
                        {
                                printf("\nExiting TPC-C Loader!\n");
                                printf("\nCheck LOGS\\ directory for
database\n");
                                printf("or table creation errors.\n");

                                // cleanup database connections and handles
                                SQLFreeHandle(SQL_HANDLE_STMT,
v_hstmt);
                                SQLDisconnect(v_hdbc);
                                SQLFreeHandle(SQL_HANDLE_DBC,
v_hdbc);

                                exit(1);
                        }
                }

                // cleanup database connections and handles
                SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
                SQLDisconnect(v_hdbc);
                SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

                return;
}
```

time.c

```
//              File:           TIME.C
//                                      Microsoft TPC-C Kit Ver. 4.21
//                                      Copyright Microsoft, 1996, 1997,
1998, 1999, 2000
//              Purpose:    Source file for time functions


// Includes
#include "tpcc.h"

// Globals
static long start_sec;
```

```
//=========================================================
================
//
// Function name: TimeNow
//
//=========================================================
================

long TimeNow()
{
  long                    time_now;
  struct      _timeb el_time;

#ifdef DEBUG
   printf("[%ld]DBG: Entering TimeNow()\n", (int) GetCurrentThreadId());
#endif

  _ftime(&el_time);

  time_now = ((el_time.time - start_sec) * 1000) + el_time.millitm;

  return time_now;
}
```

random.c

```
//              File:                           RANDOM.C
//                                              Microsoft TPC-C Kit Ver. 4.21
//                                              Copyright Microsoft, 1996, 1997,
1998, 1999, 2000
//              Purpose:    Random number generation routines for database loader


// Includes
#include "tpcc.h"
#include "math.h"

// Defines
#define A           16807
#define M       2147483647
#define Q          127773      /* M div A */
#define R            2836      /* M mod A */
#define Thread          __declspec(thread)

// Globals
long        Thread Seed = 0;        /* thread local seed */

/************************************************************************
********
*                                                                       *
* random -                                                              *
*       Implements a GOOD pseudo random number generator.  This generator      *
*       will/should? run the complete period before repeating.                *
*                                                                       *
* Copied from:                                                          *
*       Random Numbers Generators: Good Ones Are Hard to Find.                 *
*       Communications of the ACM - October 1988 Volume 31 Number 10           *
*                                                                       *
* Machine Dependencies:                                                 *
*       long must be 2 ^ 31 - 1 or greater.                             *
*                                                                       *
*************************************************************************
*******/

/************************************************************************
********
* seed - load the Seed value used in irand and drand.  Should be used before  *
*       first call to irand or drand.                                   *
*************************************************************************
*******/

void seed(long val)
{

#ifdef DEBUG
   printf("[%ld]DBG: Entering seed()...\n", (int) GetCurrentThreadId());
            printf("Old Seed %ld New Seed %ld\n",Seed, val);
#endif

   if ( val < 0 )
                val = abs(val);

   Seed = val;

}


/************************************************************************
*******
*                                                                       *
* irand - returns a 32 bit integer pseudo random number with a period of     *
*       1 to 2 ^ 32 - 1.                                                *
*                                                                       *
* parameters:                                                           *
*       none.                                                           *
*                                                                       *
* returns:                                                              *
*       32 bit integer - defined as long ( see above ).                 *
*                                                                       *
* side effects:                                                         *
*       seed get recomputed.                                            *
*************************************************************************
******/

long irand()
{
   register long    s;      /* copy of seed */
   register long    test;   /* test flag */
   register long    hi;     /* tmp value for speed */
   register long    lo;     /* tmp value for speed */

#ifdef DEBUG
   printf("[%ld]DBG: Entering irand()...\n", (int) GetCurrentThreadId());
#endif

   s = Seed;
   hi = s / Q;
   lo = s % Q;

   test = A * lo - R * hi;
   if ( test > 0 )
                        Seed = test;
   else
                        Seed = test + M;

   return( Seed );
}


/************************************************************************
*******
*                                                                       *
* drand - returns a double pseudo random number between 0.0 and 1.0.       *
*       See irand.                                                      *
*************************************************************************
******/
double drand()
{
```

```
#ifdef DEBUG
   printf("[%ld]DBG: Entering drand()...\n", (int) GetCurrentThreadId());
#endif

   return( (double)irand() / 2147483647.0);
}


//=========================================================
================
// Function   : RandomNumber
//
// Description:
//=========================================================
================
long RandomNumber(long lower, long upper)
{
   long rand_num;

#ifdef DEBUG
   printf("[%ld]DBG: Entering RandomNumber()...\n", (int)
GetCurrentThreadId());
#endif

            if ( upper == lower )   /* pgd 08-13-96 perf enhancement */
                        return lower;

            upper++;

   if ( upper <= lower )
                        rand_num = upper;
   else
                        rand_num = lower + irand() % (upper - lower); /* pgd
08-13-96 perf enhancement */


#ifdef DEBUG
   printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                                                (int) GetCurrentThreadId(), lower,
upper, rand_num);
#endif


   return rand_num;
}


#if 0

//Orginal code pgd 08/13/96

long RandomNumber(long lower,
                                                      long upper)
{
   long rand_num;

#ifdef DEBUG
   printf("[%ld]DBG: Entering RandomNumber()...\n", (int)
GetCurrentThreadId());
#endif

            upper++;

   if ((upper <= lower))
                        rand_num = upper;
   else
                        rand_num = lower + irand() % ((upper > lower) ? upper -
lower : upper);


#ifdef DEBUG
   printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                                                (int) GetCurrentThreadId(), lower,
upper, rand_num);
#endif


   return rand_num;
}
#endif

//=========================================================
================
// Function   : NURand
//
// Description:
//=========================================================
================
long NURand(int iConst,
                  long x,
                  long y,
                  long C)
{
   long rand_num;

#ifdef DEBUG
   printf("[%ld]DBG: Entering NURand()...\n", (int) GetCurrentThreadId());
#endif

   rand_num = (((RandomNumber(0,iConst) | RandomNumber(x,y)) + C) %
(y-x+1))+x;

#ifdef DEBUG
   printf("[%ld]DBG: NURand: num = %d\n", (int) GetCurrentThreadId(),
rand_num);
#endif

   return rand_num;
}
```

tpccldr.dsp

```
# Microsoft Developer Studio Project File - Name="tpccldr" - Package
Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Console Application" 0x0103

CFG=tpccldr - 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 "tpccldr.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 "tpccldr.mak" CFG="tpccldr - Win32 Release"
!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

# Begin Project
# PROP AllowPerConfigDependencies 0
```

```
# PROP Scc_ProjName ""$/mstpcc.400/setup/loader/mssql70", QROAAAAA"
# PROP Scc_LocalPath "."
CPP=cl.exe
RSC=rc.exe

!IF  "$(CFG)" == "tpccldr - 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 Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
"_CONSOLE" /YX /c
# ADD CPP /nologo /MT /W3 /GX /O2 /D "NDEBUG" /D "WIN32" /D
"_CONSOLE" /D "DBNTWIN32" /FD /c
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib
odbccp32.lib /nologo /subsystem:console /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
odbcbcp.lib /nologo /subsystem:console /pdb:none /machine:I386

!ELSEIF  "$(CFG)" == "tpccldr - 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 Ignore_Export_Lib 0
# PROP Target_Dir ""
# 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 /D "_DEBUG" /D "WIN32" /D
"_CONSOLE" /D "DBNTWIN32" /FR /FD /c
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib
odbccp32.lib /nologo /subsystem:console /debug /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
odbcbcp.lib /nologo /subsystem:console /pdb:none /debug /machine:I386

!ENDIF

# Begin Target

# Name "tpccldr - Win32 Release"
# Name "tpccldr - Win32 Debug"
# Begin Group "Source Files"

# PROP Default_Filter "cpp;c;cxx;rc;def;r;odl;hpj;bat;for;f90"
# Begin Source File

SOURCE=.\src\getargs.c
# End Source File
# Begin Source File

SOURCE=.\src\random.c
# End Source File
# Begin Source File

SOURCE=.\src\strings.c
# End Source File
# Begin Source File

SOURCE=.\src\time.c
# End Source File
# Begin Source File

SOURCE=.\src\tpccldr.c
# End Source File
# End Group
# Begin Group "Header Files"

# PROP Default_Filter "h;hpp;hxx;hm;inl;fi;fd"
# Begin Source File

SOURCE="..\..\..\..\Program Files\Microsoft Visual
Studio\VC98\Include\BASETSD.H"
# End Source File
# Begin Source File

SOURCE="C:\Program Files\Microsoft SQL
Server\80\Tools\DevTools\Include\odbcss.h"
# End Source File
# Begin Source File

SOURCE=.\src\tpcc.h
# End Source File
# End Group
# Begin Group "Resource Files"

# PROP Default_Filter "ico;cur;bmp;dlg;rc2;rct;bin;cnt;rtf;gif;jpg;jpeg;jpe"
# End Group
# End Target
# End Project
```

tpccldr.dsw

```
Microsoft Developer Studio Workspace File, Format Version 6.00
# WARNING: DO NOT EDIT OR DELETE THIS WORKSPACE FILE!

###############################################################
###############

Project: "tpccldr"=.\tpccldr.dsp - Package Owner=<4>

Package=<5>
{{{
   begin source code control
   "$/Backup/setup/loader", ZGABAAAA
   .
   end source code control
}}}

Package=<4>
{{{
}}}
```

```
#################################################################
################

Global:

Package=<5>
{{{
}}}

Package=<3>
{{{
}}}

#################################################################
################
```

tpccldr.mak

```
# Microsoft Developer Studio Generated NMAKE File, Format Version 4.10
# ** DO NOT EDIT **

# TARGTYPE "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 "tpccldr - Win32 Debug"
RSC=rc.exe
CPP=cl.exe

!IF  "$(CFG)" == "tpccldr - 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)\tpccldr.exe"

CLEAN :
	-@erase "$(INTDIR)\getargs.obj"
	-@erase "$(INTDIR)\random.obj"
	-@erase "$(INTDIR)\strings.obj"
	-@erase "$(INTDIR)\time.obj"
	-@erase "$(INTDIR)\tpccldr.obj"
	-@erase "$(OUTDIR)\tpccldr.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"$(INTDIR)/" /c
CPP_OBJS=.\objects/
CPP_SBRS=.\.
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
BSC32_FLAGS=/nologo /o"$(OUTDIR)/tpccldr.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 odbc32.lib
odbccp32.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
odbc32.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 odbc32.lib odbccp32.lib /nologo /subsystem:console /pdb:none\
 /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)
<<

!ELSEIF  "$(CFG)" == "tpccldr - 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)\tpccldr.exe"

CLEAN :
	-@erase "$(INTDIR)\getargs.obj"
	-@erase "$(INTDIR)\random.obj"
	-@erase "$(INTDIR)\strings.obj"
	-@erase "$(INTDIR)\time.obj"
	-@erase "$(INTDIR)\tpccldr.obj"
	-@erase "$(INTDIR)\vc40.idb"
	-@erase "$(INTDIR)\vc40.pdb"
	-@erase "$(OUTDIR)\tpccldr.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 /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
BSC32_FLAGS=/nologo /o"$(OUTDIR)/tpccldr.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 odbc32.lib
odbccp32.lib /nologo /subsystem:console /debug /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
odbc32.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 odbc32.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_RANDO=\
	".\src\tpcc.h"\
	"\mssql\dblib\include\sqldb.h"\
	"\mssql\dblib\include\sqlfront.h"\


"$(INTDIR)\random.obj" : $(SOURCE) $(DEP_CPP_RANDO) "$(INTDIR)"
   $(CPP) $(CPP_PROJ) $(SOURCE)


# End Source File
################################################################
################
# Begin Source File

SOURCE=.\src\strings.c
DEP_CPP_STRIN=\
	".\src\tpcc.h"\
	"\mssql\dblib\include\sqldb.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\sqldb.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\sqldb.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\sqldb.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
################################################################
################
```

Appendix C: Tunable Parameters

Microsoft Windows 2000 Advanced Server Configuration Parameters

SQL Server 2000 Stack Size

The default stack size for Microsoft SQL Server 2000 Enterprise Edition (8.00.194) was changed using the EDITBIN utility. The EDITBIN utility ships with Microsoft Visual C++ V6.0. The command used to change the stack size is:

editbin / Stack:131072

This command is fully documented as an article in the Microsoft Knowledge Base on the Microsoft Web site at www.microsoft.com/support.

Boot.ini

```
[Boot Loader]
Timeout=5
Default=
[Operating Systems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Microsoft Windows 2000
Advanced Server" /fastdetect /3gb /pae
```

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. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Returns SQL Server version string



print " "
select convert(char(30), getdate(),9)
print " "

 ------------------------------
 Nov 27 2000  9:28:52:047AM

(1 row affected)

1> 2> 3>
select @@version



 --------------------------------------------------------------------------
              ------------------------------------------------------------------------
              ------------------------------------------------------------------------
              ---------------------------------
 Microsoft SQL Server  2000 - 8.00.194 (Intel X86)
              Aug  6 2000 00:57:48
              Cop
              yright (c) 1988-2000 Microsoft Corporation
              Enterprise Edition on Windo
                                ws NT 5.0 (Build 2195: )


(1 row affected)
1> 2>
1> 2> 3> 4> 5> 6> 7> 8> 9> 10>
-- File:    CONFIG.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Collects SQL Server configuration parameters


print " "
select convert(char(30), getdate(),9)
print " "

 ------------------------------
Nov 27 2000  9:28:52:873AM

(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                                  0  2147483647          15
      15
 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         182
      182
 media retention                                0         365           0
0
 min memory per query (KB)                    512  2147483647
      512         512
```

```
min server memory (MB)                  0  2147483647
   0          0
nested triggers                         0          1          1
network packet size (B)                 512        65536
4096       4096
open objects                            0  2147483647          0
     0
priority boost                          0          1          1
query governor cost limit               0  2147483647
0          0
query wait (s)                          -1  2147483647         -1
     -1
recovery interval (min)                 0       32767         36
     36
remote access                           0          1          1

remote login timeout (s)                0  2147483647
20         20
remote proc trans                       0          1          0
0
remote query timeout (s)                0  2147483647
600        600
scan for startup procs                  0          1          0
0
set working set size                    0          1          1
1
show advanced options                   0          1          1
   1
two digit year cutoff                   1753       9999       2049
     2049
user connections                        0       32767         0
  0
user options                            0       32767         0
0

1>
```

Microsoft SQL Server 2000 Startup Parameters

c:\sqlservr -c -x -T3502 -g100

Where:

- -c Start SQL Server independently of the Service Control Manager
- -x Disables 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
- -g100 Specifies the amount of memory that is set aside for allocations not from the buffer pool

Microsoft Windows 2000 Advanced Server Configuration Parameters

System Information report written at: 11/27/2000 11:22:38 AM
[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 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	IBMSERV3
System Manufacturer	IBM
System Model	Netfinity 6000R -[8682]-
System Type	X86-based PC
Processor	x86 Family 6 Model 10 Stepping 1 GenuineIntel ~702 Mhz
Processor	x86 Family 6 Model 10 Stepping 1 GenuineIntel ~702 Mhz
Processor	x86 Family 6 Model 10 Stepping 1 GenuineIntel ~702 Mhz
Processor	x86 Family 6 Model 10 Stepping 1 GenuineIntel ~702 Mhz
BIOS Version	IBM BIOS Ver 5.0
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition1
Locale	United States
User Name	IBMSERV3\Administrator
Time Zone	Eastern Standard Time
Total Physical Memory	7,847,228 KB
Available Physical Memory	69,460 KB
Total Virtual Memory	17,651,336 KB
Available Virtual Memory	2,208,056 KB
Page File Space	9,804,108 KB
Page File	C:\pagefile.sys

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

Resource	Device
IRQ 18	S3 Inc. Savage4
IRQ 18	Adaptec AIC-7899 Ultra160/m PCI SCSI Card

[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
0x0000-0x3FFF	PCI bus	OK
0x0000-0x3FFF	Direct memory access controller	OK
0x2200-0x223F	IBM Netfinity 10/100 Ethernet Adapter	OK
0x2240-0x225F	IBM 10/100 NetFinity Fault Tolerant Adapter	OK
0x03B0-0x03BB	S3 Inc. Savage4	OK
0x03C0-0x03DF	S3 Inc. Savage4	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
0x0378-0x037F	Printer Port (LPT1)	OK
0x03F8-0x03FF	Communications Port (COM2)	OK
0x02F8-0x02FF	Communications Port (COM1)	OK
0x00E8-0x00E9	Not Available	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
0x0900-0x090F 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
0x4000-0x6FFF PCI bus OK
0x4000-0x6FFF DEC 21154 PCI to PCI bridge OK
0x4000-0x6FFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0x6000-0x60FF Adaptec AIC-7899 Ultra160/m PCI SCSI Card OK
0x6100-0x61FF Adaptec AIC-7899 Ultra160/m PCI SCSI Card OK
0x5000-0x5FFF DEC 21154 PCI to PCI bridge OK
0x5000-0x5FFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0x7000-0xFFFF PCI bus OK
0x7000-0xFFFF DEC 21154 PCI to PCI bridge OK
0x7000-0xFFFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0x9000-0x9FFF DEC 21154 PCI to PCI bridge OK
0x9000-0x9FFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0x8000-0x8FFF DEC 21154 PCI to PCI bridge OK
0x8000-0x8FFF Mylex eXtremeRAID 2000 Disk Array Controller OK

[IRQs]

IRQ Number Device
9 Microsoft ACPI-Compliant System
20 IBM Netfinity 10/100 Ethernet Adapter
16 IBM 10/100 NetFinity Fault Tolerant Adapter
18 S3 Inc. Savage4
18 Adaptec AIC-7899 Ultra160/m PCI SCSI Card
1 Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
12 PS/2 Compatible Mouse
6 Standard floppy disk controller
4 Communications Port (COM2)
3 Communications Port (COM1)
5 Not Available
8 System CMOS/real time clock
13 Numeric data processor
14 Primary IDE Channel
19 Standard OpenHCD USB Host Controller
17 Adaptec AIC-7899 Ultra160/m PCI SCSI Card
24 Mylex eXtremeRAID 2000 Disk Array Controller
25 Mylex eXtremeRAID 2000 Disk Array Controller
21 Mylex eXtremeRAID 2000 Disk Array Controller
22 Mylex eXtremeRAID 2000 Disk Array Controller
23 Mylex eXtremeRAID 2000 Disk Array Controller

[Memory]

Range Device Status
0xA0000-0xBFFFF PCI bus OK
0xA0000-0xBFFFF S3 Inc. Savage4 OK
0xEFC00000-0xF85FFFFF PCI bus OK
0xF8600000-0xFFFFFFFF PCI bus OK
0xFEBFF000-0xFEBFFFFF IBM Netfinity 10/100 Ethernet Adapter OK
0xFEA00000-0xFEAFFFFF IBM Netfinity 10/100 Ethernet Adapter OK
0xFEBFEC00-0xFEBFEC1F IBM 10/100 NetFinity Fault Tolerant Adapter OK
0xFEB00000-0xFEB7FFFF S3 Inc. Savage4 OK
0xF0000000-0xF7FFFFFF S3 Inc. Savage4 OK
0xFEBFD000-0xFEBFDFFF Standard OpenHCD USB Host Controller OK
0xE9000000-0xEA7FFFFF PCI bus OK
0xEA800000-0xEFBFFFFF PCI bus OK
0xEFBFF000-0xEFBFFFFF Adaptec AIC-7899 Ultra160/m PCI SCSI Card OK
0xEFBFE000-0xEFBFEFFF Adaptec AIC-7899 Ultra160/m PCI SCSI Card OK
0xEC000000-0xED7FFFFF DEC 21154 PCI to PCI bridge OK
0xE9800000-0xE9FFFFFF DEC 21154 PCI to PCI bridge OK
0xE9800000-0xE9FFFFFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0xED000000-0xED7FFFFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0xEE000000-0xEF7FFFFF DEC 21154 PCI to PCI bridge OK
0xEA000000-0xEA7FFFFF DEC 21154 PCI to PCI bridge OK
0xEA000000-0xEA7FFFFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0xEF000000-0xEF7FFFFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0xDF400000-0xE6BFFFFF PCI bus OK
0xE6C00000-0xE8FFFFFF PCI bus OK
0xE1000000-0xE27FFFFF DEC 21154 PCI to PCI bridge OK
0xE7800000-0xE7FFFFFF DEC 21154 PCI to PCI bridge OK
0xE7800000-0xE7FFFFFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0xE2000000-0xE27FFFFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0xE3000000-0xE47FFFFF DEC 21154 PCI to PCI bridge OK
0xE8000000-0xE87FFFFF DEC 21154 PCI to PCI bridge OK
0xE8000000-0xE87FFFFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0xE4000000-0xE47FFFFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0xE5000000-0xE67FFFFF DEC 21154 PCI to PCI bridge OK
0xE8800000-0xE8FFFFFF DEC 21154 PCI to PCI bridge OK
0xE8800000-0xE8FFFFFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0xE6000000-0xE67FFFFF 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\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
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\msg723.acm Microsoft Corporation OK
C:\WINNT\System32\MSG723.ACM 4.4.3385 106.77 KB (109,328 bytes) 11/11/2000 11:17:20 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\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) 11/11/2000 11:17:21 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

[Video Codecs]

Codec Manufacturer Description Status File Version Size Creation Date
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 6:00:00 AM
c:\winnt\system32\msh261.drv Microsoft Corporation OK C:\WINNT\System32\MSH261.DRV 4.4.3385 163.77 KB (167,696 bytes) 11/11/2000 11:17:21 AM
c:\winnt\system32\msh263.drv Microsoft Corporation OK C:\WINNT\System32\MSH263.DRV 4.4.3385 252.27 KB (258,320 bytes) 11/11/2000 11:16:46 AM
c:\winnt\system32\msvidc32.dll Microsoft Corporation OK C:\WINNT\System32\MSVIDC32.DLL 5.00.2134.1 27.27 KB (27,920 bytes) 12/7/1999 6:00:00 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 6:00:00 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 6:00:00 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 6:00:00 AM

[CD-ROM]

Item Value
Drive D:
Description CD-ROM Drive
Media Loaded False
Media Type CD-ROM
Name LITEON CD-ROM LTN403
Manufacturer (Standard CD-ROM drives)
Status OK
Transfer Rate Not Available
SCSI Target ID 0
PNP Device ID IDE\CDROMLITEON_CD-ROM_LTN403_____DU26___\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_04\3&267A616A&0&30
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 800 x 600 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&F0B8F99&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&F0B8F99&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] IBM 10/100 NetFinity Fault Tolerant Adapter
Adapter Type Ethernet 802.3
Product Name IBM 10/100 NetFinity Fault Tolerant Adapter
Installed True
PNP Device ID PCI\VEN_1022&DEV_2000&SUBSYS_20001014&REV_43\3&267A616A&0&28
Last Reset 11/27/2000 5:40:17 AM
Index 0
Service Name PCnet
IP Address 9.67.179.220
IP Subnet 255.255.255.192
Default IP Gateway 9.67.179.193
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:06:29:F6:02:BD

Service Name PCnet
IRQ Number 16
I/O Port 0x2240-0x225F
Driver c:\winnt\system32\drivers\pcntn5m.sys (33811, 4.23.00)

Name [00000001] IBM Netfinity 10/100 Ethernet Adapter
Adapter Type Ethernet 802.3
Product Name IBM Netfinity 10/100 Ethernet Adapter
Installed True
PNP Device ID
PCI\VEN_8086&DEV_1229&SUBSYS_105C1014&REV_08\3&267A616A&0&08
Last Reset 11/27/2000 5:40:17 AM
Index 1
Service Name IBMFE
IP Address 192.168.125.254
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 00:06:29:6F:8F:FC
Service Name IBMFE
IRQ Number 20
I/O Port 0x2200-0x223F
Driver c:\winnt\system32\drivers\ibmfent5.sys (80656, 4.01.75.0000)

Name [00000002] RAS Async Adapter
Adapter Type Not Available
Product Name RAS Async Adapter
Installed True
PNP Device ID Not Available
Last Reset 11/27/2000 5:40:17 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 11/27/2000 5:40:17 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 11/27/2000 5:40:17 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\raspptp.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_PTIMINIPORT\0000
Last Reset 11/27/2000 5:40:17 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 11/27/2000 5:40:17 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 (90768, 5.00.2184.1)

[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_{B1D8F20D-6ACF-456A-BA9B-9F1E8F640427}] SEQPACKET 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_{B1D8F20D-6ACF-456A-BA9B-9F1E8F640427}] 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_{59EFE692-FDBA-4E4E-BF99-9BACB8A7A8F0}] 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_{59EFE692-FDBA-4E4E-BF99-9BACB8A7A8F0}] 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_{5361725E-E64D-41CF-8C7F-C49E4994CA9A}] 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_{5361725E-E64D-41CF-8C7F-C49E4994CA9A}] 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_{6F9AA8C4-A0D3-41E1-BAD0-42C67E9585CF}] 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_{6F9AA8C4-A0D3-41E1-BAD0-42C67E9585CF}] 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\wsock32.dll
Version	5.00.2152.1
Size	21.27 KB (21,776 bytes)

[Ports]

[Following are sub-categories of this main category]

[Serial]

Item	Value
Name	COM2
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 (62448, 5.00.2134.1)

Name	COM1
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 (62448, 5.00.2134.1)

[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 4.59 GB (4,926,988,288 bytes)
Volume Name
Volume Serial Number 18FF945A
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 SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 3
Drive SCSITargetId 2
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
Partition Disk #2, Partition #0
Partition Size 37.81 GB (40,599,949,824 bytes)
Starting Offset 32256 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 SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 5
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 471974791680 bytes
Drive TotalCylinders 57381
Drive TotalSectors 921825765
Drive TotalTracks 14632155
Drive TracksPerCylinder 255

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
Partition Disk #2, Partition #1
Partition Size 17.63 GB (18,926,369,280 bytes)
Starting Offset Not Available
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 SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 5
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 471974791680 bytes
Drive TotalCylinders 57381
Drive TotalSectors 921825765

Drive TotalTracks 14632155
Drive TracksPerCylinder 255

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
Partition Disk #3, Partition #0
Partition Size 37.81 GB (40,599,949,824 bytes)
Starting Offset 32256 bytes
Drive Description \\.\PHYSICALDRIVE3
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 6
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 471974791680 bytes
Drive TotalCylinders 57381
Drive TotalSectors 921825765
Drive TotalTracks 14632155
Drive TracksPerCylinder 255

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
Partition Disk #3, Partition #1
Partition Size 17.63 GB (18,926,369,280 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALDRIVE3
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 6
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 471974791680 bytes
Drive TotalCylinders 57381
Drive TotalSectors 921825765
Drive TotalTracks 14632155
Drive TracksPerCylinder 255

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
Partition Disk #4, Partition #0
Partition Size 37.81 GB (40,599,949,824 bytes)
Starting Offset 32256 bytes
Drive Description \\.\PHYSICALDRIVE4
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 3
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 7
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 471974791680 bytes
Drive TotalCylinders 57381
Drive TotalSectors 921825765
Drive TotalTracks 14632155
Drive TracksPerCylinder 255

Drive J:
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
Partition Disk #4, Partition #1
Partition Size 17.63 GB (18,926,369,280 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALDRIVE4
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 3
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 7
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 471974791680 bytes
Drive TotalCylinders 57381
Drive TotalSectors 921825765
Drive TotalTracks 14632155
Drive TracksPerCylinder 255

Drive K:
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
Partition Disk #5, Partition #0
Partition Size 37.81 GB (40,599,949,824 bytes)
Starting Offset 32256 bytes
Drive Description \\.\PHYSICALDRIVE5
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 3
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 8
Drive SCSITargetId 0
Drive SectorsPerTrack 63

Drive Size 945257402880 bytes
Drive TotalCylinders 114921
Drive TotalSectors 1846205865
Drive TotalTracks 29304855
Drive TracksPerCylinder 255

Drive L:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #5, Partition #1
Partition Size 17.63 GB (18,926,369,280 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALDRIVE5
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 3
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 8
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 945257402880 bytes
Drive TotalCylinders 114921
Drive TotalSectors 1846205865
Drive TotalTracks 29304855
Drive TracksPerCylinder 255

Drive O:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #1, Partition #0
Partition Size 83.01 GB (89,129,101,824 bytes)
Starting Offset 32256 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 1
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 4
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 109067212800 bytes
Drive TotalCylinders 13260
Drive TotalSectors 213021900
Drive TotalTracks 3381300
Drive TracksPerCylinder 255

Drive Y:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 117.19 GB (125,830,332,416 bytes)
Free Space 17.35 GB (18,630,217,728 bytes)
Volume Name Backup1&2
Volume Serial Number 4CBDE553
Partition Disk #4, Partition #2
Partition Size 117.19 GB (125,830,333,440 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALDRIVE4
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 3
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 7
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 471974791680 bytes
Drive TotalCylinders 57381
Drive TotalSectors 921825765
Drive TotalTracks 14632155
Drive TracksPerCylinder 255

Drive Z:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 117.19 GB (125,830,332,416 bytes)
Free Space 17.33 GB (18,613,080,064 bytes)
Volume Name Backup3&4
Volume Serial Number ACD729B8
Partition Disk #5, Partition #2
Partition Size 117.19 GB (125,830,333,440 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALDRIVE5
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 3
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 8
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 945257402880 bytes
Drive TotalCylinders 114921
Drive TotalSectors 1846205865
Drive TotalTracks 29304855
Drive TracksPerCylinder 255

[SCSI]

Item Value
Name Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Caption Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Driver adpu160m
Status OK
PNP Device ID
PCI\VEN_9005&DEV_00CF&SUBSYS_00CF9005&REV_01\3&1070020&0&08
Device ID
PCI\VEN_9005&DEV_00CF&SUBSYS_00CF9005&REV_01\3&1070020&0&08
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 17
I/O Port 0x6000-0x60FF
Driver c:\winnt\system32\drivers\adpu160m.sys (64432, v3.10a)

Name Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Caption Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Driver adpu160m
Status OK
PNP Device ID
PCI\VEN_9005&DEV_00CF&SUBSYS_00CF9005&REV_01\3&1070020&0&09
Device ID
PCI\VEN_9005&DEV_00CF&SUBSYS_00CF9005&REV_01\3&1070020&0&09
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 18
I/O Port 0x6100-0x61FF
Driver c:\winnt\system32\drivers\adpu160m.sys (64432, v3.10a)

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&375C4928&0&4028
Device ID
PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&375C4928&0&4028
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 24
I/O Port 0x5000-0x5FFF
Driver c:\winnt\system32\drivers\dac2w2k.sys (185520, 9.00-03)

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&18534677&0&4030
Device ID
PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&18534677&0&4030
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 25
I/O Port 0x4000-0x6FFF
Driver c:\winnt\system32\drivers\dac2w2k.sys (185520, 9.00-03)

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&372A30F3&0&4010
Device ID
PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&372A30F3&0&4010
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 21
I/O Port 0x9000-0x9FFF
Driver c:\winnt\system32\drivers\dac2w2k.sys (185520, 9.00-03)

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&8928022&0&4018
Device ID
PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&8928022&0&4018
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 22
I/O Port 0x8000-0x8FFF
Driver c:\winnt\system32\drivers\dac2w2k.sys (185520, 9.00-03)

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&279B82D3&0&4020
Device ID
PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&279B82D3&0&4020
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 23
I/O Port 0x7000-0xFFFF
Driver c:\winnt\system32\drivers\dac2w2k.sys (185520, 9.00-03)

[Printing]

Name Port Name Server Name
No printing information

[Problem Devices]

Device PNP Device ID Error Code
Not Available ACPI\IBM37D0\4&F0B8F99&0 28
Not Available ACPI\IBM37C0\4&F0B8F99&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 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
asyncmac    RAS Asynchronous Media Driver
c:\winnt\system32\drivers\asyncmac.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
dac2w2k    dac2w2k    c:\winnt\system32\drivers\dac2w2k.sys    Kernel
Driver    True    Boot    Running    OK    Normal    False
True
dac960nt    dac960nt    Not Available    Kernel Driver    False
Disabled    Stopped    OK    Normal    False    False
deckzpsx    deckzpsx    Not Available    Kernel Driver    False
Disabled    Stopped    OK    Normal    False    False
dfsdriver    DfsDriver    c:\winnt\system32\drivers\dfs.sys    File System Driver
True    Boot    Running    OK    Normal    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    Kernel
Driver    False    Disabled    Stopped    OK    Normal    False
False
dmboot    dmboot    c:\winnt\system32\drivers\dmboot.sys    Kernel
Driver    False    Disabled    Stopped    OK    Normal    False
False
dmio    Logical Disk Manager Driver    c:\winnt\system32\drivers\dmio.sys
Kernel Driver    True    Boot    Running    OK    Normal
False    True
dmload    dmload    c:\winnt\system32\drivers\dmload.sys    Kernel
Driver    True    Boot    Running    OK    Normal    False
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    File System
Driver    True    Disabled    Running    OK    Normal    False
True
fd16_700    Fd16_700    Not Available    Kernel Driver    False
Disabled    Stopped    OK    Normal    False    False
fdc    Floppy Disk Controller Driver    c:\winnt\system32\drivers\fdc.sys
Kernel Driver    True    Manual    Running    OK    Normal
False    True
fireport    fireport    Not Available    Kernel Driver    False
Disabled    Stopped    OK    Normal    False    False
flashpnt    flashpnt    Not Available    Kernel Driver    False
Disabled    Stopped    OK    Normal    False    False
flpydisk    Floppy Disk Driver    c:\winnt\system32\drivers\flpydisk.sys
Kernel Driver    True    Manual    Running    OK    Normal
False    True
ftdisk    Volume Manager Driver    c:\winnt\system32\drivers\ftdisk.sys
Kernel Driver    True    Boot    Running    OK    Normal
False    True
gamdrv    gamdrv    c:\winnt\system32\drivers\gamdrv.sys    Kernel
Driver    True    Boot    Running    OK    Normal    False
True
gpc    Generic Packet Classifier    c:\winnt\system32\drivers\msgpc.sys
Kernel Driver    True    Manual    Running    OK    Normal
False    True
i8042prt    i8042 Keyboard and PS/2 Mouse Port Driver
c:\winnt\system32\drivers\i8042prt.sys    Kernel Driver    True
System    Running    OK    Normal    False    True
ibmfe    IBM 10/100 Ethernet PCI Adapter Driver
c:\winnt\system32\drivers\ibmfent5.sys    Kernel Driver    True
Manual    Running    OK    Normal    False    True
ini910u    ini910u    Not Available    Kernel Driver    False
Disabled    Stopped    OK    Normal    False    False
intelide    IntelIde    Not Available    Kernel Driver    False
Disabled    Stopped    OK    Normal    False    False
ipfilterdriver    IP Traffic Filter Driver
c:\winnt\system32\drivers\ipfltdrv.sys    Kernel Driver    False
Manual    Stopped    OK    Normal    False    False
ipinip    IP in IP Tunnel Driver    c:\winnt\system32\drivers\ipinip.sys
Kernel Driver    False    Manual    Stopped    OK    Normal
False    False
ipnat    IP Network Address Translator    c:\winnt\system32\drivers\ipnat.sys
Kernel Driver    False    Manual    Stopped    OK    Normal
False    False
ipsec    IPSEC driver    c:\winnt\system32\drivers\ipsec.sys
Kernel Driver    True    Manual    Running    OK    Normal
False    True
ipsraidn    ipsraidn    Not Available    Kernel Driver    False
Disabled    Stopped    OK    Normal    False    False
isapnp    PnP ISA/EISA Bus Driver    c:\winnt\system32\drivers\isapnp.sys
Kernel Driver    True    Boot    Running    OK    Critical
False    True
kbdclass    Keyboard Class Driver    c:\winnt\system32\drivers\kbdclass.sys
Kernel Driver    True    System    Running    OK    Normal
False    True
```

```
ksecdd KSecDD c:\winnt\system32\drivers\ksecdd.sys Kernel
Driver True Boot Running OK Normal False
True
lbrtfdc lbrtfdc Not Available Kernel Driver False
System Stopped OK Ignore False False
lp6nds35 lp6nds35 Not Available Kernel Driver False
Disabled Stopped OK Normal False False
macdisk macdisk c:\winnt\system32\drivers\mac2w2k.sys Kernel
Driver True Boot Running OK Normal False
True
mnmdd mnmdd c:\winnt\system32\drivers\mnmdd.sys Kernel
Driver True System Running OK Ignore False
True
modem Modem c:\winnt\system32\drivers\modem.sys Kernel
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
False True
mountmgr MountMgr c:\winnt\system32\drivers\mountmgr.sys Kernel
Driver True Boot Running OK Normal False
True
mraid35x mraid35x Not Available Kernel Driver False
Disabled Stopped OK Normal False False
mrxsmb MRXSMB c:\winnt\system32\drivers\mrxsmb.sys File System
Driver True System Running OK Normal False
True
msfs Msfs c:\winnt\system32\drivers\msfs.sys File System Driver
True System Running OK Normal False True
mskssrv Microsoft Streaming Service Proxy
c:\winnt\system32\drivers\mskssrv.sys Kernel Driver False
Manual Stopped OK Normal False False
mspclock Microsoft Streaming Clock Proxy
c:\winnt\system32\drivers\mspclock.sys Kernel Driver False
Manual Stopped OK Normal False False
mspqm Microsoft Streaming Quality Manager Proxy
c:\winnt\system32\drivers\mspqm.sys Kernel Driver False
Manual Stopped OK Normal False False
mup Mup c:\winnt\system32\drivers\mup.sys File System Driver
True Boot Running OK Normal False True
ncrc710 Ncrc710 Not Available Kernel Driver False
Disabled Stopped OK Normal False False
ndis NDIS System Driver c:\winnt\system32\drivers\ndis.sys Kernel
Driver True Boot Running OK Normal False
True
ndistapi Remote Access NDIS TAPI Driver
c:\winnt\system32\drivers\ndistapi.sys Kernel Driver True
Manual Running OK Normal False True
ndiswan Remote Access NDIS WAN Driver
c:\winnt\system32\drivers\ndiswan.sys Kernel Driver True
Manual Running OK Normal False True
ndproxy NDIS Proxy c:\winnt\system32\drivers\ndproxy.sys
Kernel Driver True Manual Running OK Normal
False True
netbios NetBIOS Interface c:\winnt\system32\drivers\netbios.sys
File System Driver True System Running OK Normal
False True
netbt NetBios over Tcpip c:\winnt\system32\drivers\netbt.sys
Kernel Driver True System Running OK Normal
False True
netdetect NetDetect c:\winnt\system32\drivers\netdtect.sys Kernel
Driver False Manual Stopped OK Normal False
False
npfs Npfs c:\winnt\system32\drivers\npfs.sys File System Driver
True System Running OK Normal False True
ntfs Ntfs c:\winnt\system32\drivers\ntfs.sys File System Driver
True Disabled Running OK Normal False True
null Null c:\winnt\system32\drivers\null.sys Kernel Driver
True System Running OK Normal False True
nwlnkflt IPX Traffic Filter Driver
c:\winnt\system32\drivers\nwlnkflt.sys Kernel Driver False
Manual Stopped OK Normal False False
nwlnkfwd IPX Traffic Forwarder Driver
c:\winnt\system32\drivers\nwlnkfwd.sys Kernel Driver False
Manual Stopped OK Normal False False
openhci Microsoft USB Open Host Controller Driver
c:\winnt\system32\drivers\openhci.sys Kernel Driver True
Manual Running OK Normal False True
parallel Parallel class driver c:\winnt\system32\drivers\parallel.sys
Kernel Driver True Manual Running OK Normal
False True
parport Parallel port driver c:\winnt\system32\drivers\parport.sys
Kernel Driver True System Running OK Ignore
False True
partmgr PartMgr c:\winnt\system32\drivers\partmgr.sys Kernel
Driver True Boot Running OK Normal False
True
parvdm ParVdm c:\winnt\system32\drivers\parvdm.sys Kernel
Driver True Auto Running OK Ignore False
True
pci PCI Bus Driver c:\winnt\system32\drivers\pci.sys Kernel
Driver True Boot Running OK Critical False
True
pcidump PCIDump Not Available Kernel Driver False
System Stopped OK Ignore False False
pciide PCIIde c:\winnt\system32\drivers\pciide.sys Kernel
Driver True Boot Running OK Normal False
True
pcmcia Pcmcia c:\winnt\system32\drivers\pcmcia.sys Kernel
Driver False Disabled Stopped OK Normal False
False
pcnet PCNET Adapter Driver
c:\winnt\system32\drivers\pcntn5m.sys Kernel Driver True
Manual Running OK Normal False True
pdcomp PDCOMP Not Available Kernel Driver False
Manual Stopped OK Ignore False False
pdframe PDFRAME Not Available Kernel Driver
False Manual Stopped OK Ignore False False
pdreli PDRELI Not Available Kernel Driver False
Manual Stopped OK Ignore False False
pdrframe PDRFRAME Not Available Kernel Driver
False Manual Stopped OK Ignore False False
pptpminiport WAN Miniport (PPTP)
c:\winnt\system32\drivers\raspptp.sys Kernel Driver True
Manual Running OK Normal False True
ptilink Direct Parallel Link Driver c:\winnt\system32\drivers\ptilink.sys
Kernel Driver True Manual Running OK Normal
False True
ql1080 ql1080 Not Available Kernel Driver False
Disabled Stopped OK Normal False False
ql10wnt Ql10wnt Not Available Kernel Driver False
Disabled Stopped OK Normal False False
ql1240 ql1240 Not Available Kernel Driver False
Disabled Stopped OK Normal False False
ql2100 ql2100 Not Available Kernel Driver False
Disabled Stopped OK Normal False False
rasacd Remote Access Auto Connection Driver
c:\winnt\system32\drivers\rasacd.sys Kernel Driver True
System Running OK Normal False True
rasl2tp WAN Miniport (L2TP)
c:\winnt\system32\drivers\rasl2tp.sys Kernel Driver True
Manual Running OK Normal False True
raspti Direct Parallel c:\winnt\system32\drivers\raspti.sys
Kernel Driver True Manual Running OK Normal
False True
rca Microsoft Streaming Network Raw Channel Access
c:\winnt\system32\drivers\rca.sys Kernel Driver False Manual
Stopped OK Normal False False
```

rdbss Rdbss c:\winnt\system32\drivers\rdbss.sys File System Driver True System Running OK Normal False True
rdpwd RDPWD c:\winnt\system32\drivers\rdpwd.sys Kernel Driver False Manual Stopped OK Ignore False False
redbook Digital CD Audio Playback Filter Driver c:\winnt\system32\drivers\redbook.sys Kernel Driver False System Stopped OK Normal False False
s3inc S3Inc c:\winnt\system32\drivers\s3sav4m.sys Kernel Driver True Manual Running OK Ignore False True
serenum Serenum Filter Driver c:\winnt\system32\drivers\serenum.sys Kernel Driver True Manual Running OK Normal False True
serial Serial port driver c:\winnt\system32\drivers\serial.sys Kernel Driver True System Running OK Ignore False True
sfloppy Sfloppy c:\winnt\system32\drivers\sfloppy.sys Kernel Driver False System Stopped OK Ignore False False
sglfb sglfb Not Available Kernel Driver False System Stopped OK Normal False False
simbad Simbad Not Available Kernel Driver False Disabled Stopped OK Normal False False
sparrow Sparrow Not Available Kernel Driver False Disabled Stopped OK Normal False False
spud Special Purpose Utility Driver c:\winnt\system32\drivers\spud.sys Kernel Driver False Manual Stopped OK Normal False False
srv Srv c:\winnt\system32\drivers\srv.sys File System Driver True Manual Running OK Normal False True
swenum Software Bus Driver c:\winnt\system32\drivers\swenum.sys Kernel Driver True Manual Running OK Normal False True
symc810 symc810 Not Available Kernel Driver False Disabled Stopped OK Normal False False
symc8xx symc8xx Not Available Kernel Driver False Disabled Stopped OK Normal False False
sym_hi sym_hi Not Available Kernel Driver False Disabled Stopped OK Normal False False
tcpip TCP/IP Protocol Driver c:\winnt\system32\drivers\tcpip.sys Kernel Driver True System Running OK Normal False True
tdasync TDASYNC c:\winnt\system32\drivers\tdasync.sys Kernel Driver False Manual Stopped OK Ignore False False
tdipx TDIPX c:\winnt\system32\drivers\tdipx.sys Kernel Driver False Manual Stopped OK Ignore False False
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
tdspx TDSPX c:\winnt\system32\drivers\tdspx.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 False 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 False 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 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 False Manual Stopped OK Ignore False False

[Environment Variables]

Variable Value User Name
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
NUMBER_OF_PROCESSORS 4 <SYSTEM>
OS Windows_NT <SYSTEM>
Os2LibPath %SystemRoot%\system32\os2\dll; <SYSTEM>
Path %SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\Program Files\Microsoft SQL Server\80\Tools\BINN;C:\Program Files\Microsoft SQL Server\MSSQL\BINN <SYSTEM>
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH <SYSTEM>
PROCESSOR_ARCHITECTURE x86 <SYSTEM>
PROCESSOR_IDENTIFIER x86 Family 6 Model 10 Stepping 1, GenuineIntel <SYSTEM>
PROCESSOR_LEVEL 6 <SYSTEM>
PROCESSOR_REVISION 0a01 <SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>
windir %SystemRoot% <SYSTEM>
TEMP %USERPROFILE%\Local Settings\Temp IBMSERV3\Administrator
TMP %USERPROFILE%\Local Settings\Temp IBMSERV3\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 Unknown Unknown

[Network Connections]

Local Name Remote Name Type Status User Name
M: \\9.37.221.197\c$ Disk Error

[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 168 11 204800 1413120 11/27/2000 10:40:53 AM 5.00.2170.1 44.27 KB (45,328 bytes) 12/7/1999 6:00:00 AM

csrss.exe Not Available 196 13 Not Available Not Available 11/27/2000 10:40:58 AM Unknown Unknown Unknown
winlogon.exe c:\winnt\system32\winlogon.exe 216 13 204800 1413120 11/27/2000 10:41:00 AM 5.00.2182.1 173.27 KB (177,424 bytes) 12/7/1999 6:00:00 AM
services.exe c:\winnt\system32\services.exe 244 9 204800 1413120 11/27/2000 10:41:01 AM 5.00.2134.1 86.77 KB (88,848 bytes) 12/7/1999 6:00:00 AM
lsass.exe c:\winnt\system32\lsass.exe 256 13 204800 1413120 11/27/2000 10:41:01 AM 5.00.2184.1 32.77 KB (33,552 bytes) 12/7/1999 6:00:00 AM
svchost.exe c:\winnt\system32\svchost.exe 384 8 204800 1413120 11/27/2000 10:41:04 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 408 8 204800 1413120 11/27/2000 10:41:05 AM 1999.9.3421.3 6.77 KB (6,928 bytes) 11/11/2000 6:11:27 AM
tcpsvcs.exe c:\winnt\system32\tcpsvcs.exe 612 8 204800 1413120 11/27/2000 10:41:07 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 632 8 204800 1413120 11/27/2000 10:41:07 AM 1.50.1085.0001 188.05 KB (192,567 bytes) 12/7/1999 6:00:00 AM
svchost.exe c:\winnt\system32\svchost.exe 596 8 204800 1413120 11/27/2000 10:48:01 AM 5.00.2134.1 7.77 KB (7,952 bytes) 12/7/1999 6:00:00 AM
explorer.exe c:\winnt\explorer.exe 884 8 204800 1413120 11/27/2000 10:48:01 AM 5.00.2920.0000 232.77 KB (238,352 bytes) 12/7/1999 6:00:00 AM
ibmmon.exe c:\winnt\system32\ibmmon.exe 964 8 204800 1413120 11/27/2000 10:48:03 AM 1.11 28.50 KB (29,184 bytes) 11/12/2000 8:29:48 AM
sqlmangr.exe c:\program files\microsoft sql server\80\tools\binn\sqlmangr.exe 948 8 204800 1413120 11/27/2000 10:48:04 AM 2000.080.0194.00 68.00 KB (69,632 bytes) 11/12/2000 8:48:51 AM
cmd.exe c:\winnt\system32\cmd.exe 908 8 204800 1413120 11/27/2000 10:48:10 AM 5.00.2144.1 230.77 KB (236,304 bytes) 12/7/1999 6:00:00 AM
cmd.exe c:\winnt\system32\cmd.exe 396 8 204800 1413120 11/27/2000 11:05:55 AM 5.00.2144.1 230.77 KB (236,304 bytes) 12/7/1999 6:00:00 AM
sqlservr_194_s2 c:\program files\microsoft sql server\mssql\binn\sqlservr_194_s2.exe 968 13 204800 1413120 11/27/2000 11:11:55 AM 2000.080.0194.00 7.10 MB (7,442,493 bytes) 11/12/2000 9:02:16 AM
gamscm.exe c:\winnt\system32\gamserv\gamscm.exe 1000 8 204800 1413120 11/27/2000 11:13:51 AM Not Available 119.28 KB (122,144 bytes) 11/12/2000 9:13:29 AM
gamserv.exe c:\winnt\system32\gamserv\gamserv.exe 1012 13 204800 1413120 11/27/2000 11:13:51 AM Not Available 128.70 KB (131,793 bytes) 11/12/2000 9:13:29 AM
gamevent.exe c:\winnt\system32\gamserv\gamevent.exe 1020 13 204800 1413120 11/27/2000 11:13:51 AM Not Available 88.71 KB (90,834 bytes) 11/12/2000 9:13:29 AM
gamevlog.exe c:\winnt\system32\gamserv\gamevlog.exe 1028 13 204800 1413120 11/27/2000 11:13:51 AM Not Available 187.35 KB (191,842 bytes) 11/12/2000 9:13:29 AM
isqlw.exe c:\program files\microsoft sql server\80\tools\binn\isqlw.exe 832 8 204800 1413120 11/27/2000 11:18:16 AM 2000.080.0194.00 344.06 KB (352,319 bytes) 11/12/2000 8:48:34 AM
mmc.exe c:\winnt\system32\mmc.exe 1080 8 204800 1413120 11/27/2000 11:20:21 AM 5.00.2153.1 589.27 KB (603,408 bytes) 12/7/1999 6:00:00 AM
rsvp.exe c:\winnt\system32\rsvp.exe 1232 8 204800 1413120 11/27/2000 11:21:45 AM 5.00.2167.1 172.77 KB (176,912 bytes) 12/7/1999 6:00:00 AM
mdm.exe c:\winnt\system32\mdm.exe 1152 8 204800 1413120 11/27/2000 11:22:00 AM 6.00.8424 121.29 KB (124,200 bytes) 11/11/2000 6:14:31 AM

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
mdm.exe	6.00.8424	121.29 KB (124,200 bytes)	11/11/2000 6:14:31 AM	Microsoft Corporation	c:\winnt\system32\mdm.exe
traffic.dll	5.00.2139.1	30.77 KB (31,504 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\traffic.dll
rsvp.exe	5.00.2167.1	172.77 KB (176,912 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\rsvp.exe
wbemprox.dll	1.50.1085.0001	40.05 KB (41,016 bytes)	12/7/1999 6:00:00 AM	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	5.00.2191.1	194.27 KB (198,928 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\adsnt.dll
dbghelp.dll	5.00.2195.1	159.27 KB (163,088 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\dbghelp.dll
localsec.dll	5.00.2134.1	227.27 KB (232,720 bytes)	12/7/1999 6:00:00 AM	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
filemgmt.dll	5.00.2134.1	287.27 KB (294,160 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\filemgmt.dll
pdh.dll	5.00.2174.1	143.27 KB (146,704 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\pdh.dll
smlogcfg.dll	5.00.2163.1	273.27 KB (279,824 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\smlogcfg.dll
cabinet.dll	5.00.2147.1	54.77 KB (56,080 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\cabinet.dll
msinfo32.dll	5.00.2177.1	312.27 KB (319,760 bytes)	11/11/2000 11:17:16 AM	Microsoft Corporation	c:\program files\common files\microsoft shared\msinfo\msinfo32.dll
riched20.dll	5.30.23.1200	421.27 KB (431,376 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\riched20.dll
riched32.dll	5.00.2134.1	3.77 KB (3,856 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\riched32.dll
els.dll	5.00.2175.1	151.27 KB (154,896 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\els.dll
ntmsmgr.dll	1,0,0,1	427.77 KB (438,032 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation and HighGround Systems, Inc.	c:\winnt\system32\ntmsmgr.dll
mmfutil.dll	1.50.1085.0000	32.06 KB (32,829 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\mmfutil.dll
logdrive.dll	1.50.1085.0000	200.06 KB (204,863 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\logdrive.dll
dfrgres.dll	5.00.2150.1	27.50 KB (28,160 bytes)	12/7/1999 6:00:00 AM	Executive Software International, Inc.	c:\winnt\system32\dfrgres.dll
dfrgsnap.dll	5.00.2150.1	41.77 KB (42,768 bytes)	12/7/1999 6:00:00 AM	Executive Software International, Inc.	c:\winnt\system32\dfrgsnap.dll
dmdskres.dll	2191.1.296.2	119.00 KB (121,856 bytes)	12/7/1999 6:00:00 AM	Microsoft Corp., VERITAS Software	c:\winnt\system32\dmdskres.dll
dmutil.dll	2191.1.296.2	41.77 KB (42,768 bytes)	12/7/1999 6:00:00 AM	VERITAS Software Corp.	c:\winnt\system32\dmutil.dll
ntmsapi.dll	5.00.1948.1	50.27 KB (51,472 bytes)	12/7/1999 6:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntmsapi.dll
dmdskmgr.dll	2191.1.296.2	158.77 KB (162,576 bytes)	12/7/1999 6:00:00 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 6:00:00 AM Microsoft Corporation c:\winnt\system32\mycomput.dll
mmcndmgr.dll 5.00.2178.1 815.27 KB (834,832 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\mmcndmgr.dll
mmc.exe 5.00.2153.1 589.27 KB (603,408 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\mmc.exe
dbmslpcn.dll 2000.080.0194.00 28.06 KB (28,734 bytes) 11/12/2000 8:48:08 AM Microsoft Corporation c:\winnt\system32\dbmslpcn.dll
dbnetlib.dll2000.080.0194.00 84.06 KB (86,082 bytes) 11/12/2000 8:47:27 AM Microsoft Corporationc:\winnt\system32\dbnetlib.dll
sqllex.dll 2000.080.0194.00 148.06 KB (151,616 bytes) 11/12/2000 8:48:50 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\sqllex.dll
objmgr.rll 2000.080.0194.00 56.00 KB (57,344 bytes) 11/12/2000 8:48:35 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\resources\1033\objmgr.rll
objmgr.dll 2000.080.0194.00 308.06 KB (315,456 bytes) 11/12/2000 8:48:35 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\objmgr.dll
odbccp32.dll 3.520.6526.0 100.27 KB (102,672 bytes) 11/12/2000 8:47:23 AM Microsoft Corporation c:\winnt\system32\odbccp32.dll
sqlsrv32.rll 2000.080.0194.00 88.00 KB (90,112 bytes) 11/12/2000 8:47:28 AM Microsoft Corporationc:\winnt\system32\sqlsrv32.rll
sqlsrv32.dll 2000.080.0194.00 460.08 KB (471,119 bytes) 11/12/2000 8:47:28 AM Microsoft Corporation c:\winnt\system32\sqlsrv32.dll
isqlw.rll 2000.080.0194.00 240.00 KB (245,760 bytes) 11/12/2000 8:48:35 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\resources\1033\isqlw.rll
sqlqry.rll 2000.080.0194.00 180.00 KB (184,320 bytes) 11/12/2000 8:48:35 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\resources\1033\sqlqry.rll
pfutil80.rll 2000.080.0194.00 144.00 KB (147,456 bytes) 11/12/2000 8:48:50 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\resources\1033\pfutil80.rll
pfclnt80.rll2000.080.0194.00 28.00 KB (28,672 bytes) 11/12/2000 8:48:48 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\resources\1033\pfclnt80.rll
semsfc.rll 2000.080.0194.00 24.00 KB (24,576 bytes) 11/12/2000 8:48:50 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\resources\1033\semsfc.rll
sqlgui.rll 2000.080.0194.00 56.00 KB (57,344 bytes) 11/12/2000 8:48:50 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\resources\1033\sqlgui.rll
pfclnt80.dll 2000.080.0194.00 404.06 KB (413,762 bytes) 11/12/2000 8:48:48 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\pfclnt80.dll
semsfc.dll 2000.080.0194.00 224.06 KB (229,440 bytes) 11/12/2000 8:48:48 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\semsfc.dll
pfutil80.dll2000.080.0194.00 268.06 KB (274,498 bytes) 11/12/2000 8:48:50 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\pfutil80.dll
sqlqry.dll 2000.080.0194.00 392.06 KB (401,472 bytes) 11/12/2000 8:48:35 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\sqlqry.dll
imm32.dll 5.00.2180.1 93.77 KB (96,016 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\imm32.dll
sqlgui.dll 2000.080.0194.00 444.06 KB (454,720 bytes) 11/12/2000 8:48:48 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\sqlgui.dll
isqlw.exe 2000.080.0194.00 344.06 KB (352,319 bytes) 11/12/2000 8:48:34 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\isqlw.exe
gamevlog.exe Not Available 187.35 KB (191,842 bytes) 11/12/2000 9:13:29 AM Not Available c:\winnt\system32\gamserv\gamevlog.exe
gamevent.exe Not Available 88.71 KB (90,834 bytes) 11/12/2000 9:13:29 AM Not Available c:\winnt\system32\gamserv\gamevent.exe
gamserv.exe Not Available 128.70 KB (131,793 bytes) 11/12/2000 9:13:29 AM Not Available c:\winnt\system32\gamserv\gamserv.exe
gamscm.exe Not Available 119.28 KB (122,144 bytes) 11/12/2000 9:13:29 AM Not Available c:\winnt\system32\gamserv\gamscm.exe
oledb32r.dll 2.60.6526.0 68.27 KB (69,904 bytes) 11/12/2000 8:47:24 AM Microsoft Corporationc:\program files\common files\system\ole db\oledb32r.dll
oledb32.dll2.60.6526.0 448.27 KB (459,024 bytes) 11/12/2000 8:47:24 AM Microsoft Corporationc:\program files\common files\system\ole db\oledb32.dll
msdatl3.dll2.60.6526.0 92.27 KB (94,480 bytes) 11/12/2000 8:47:22 AM Microsoft Corporationc:\program files\common files\system\ole db\msdatl3.dll
msdart.dll 2.60.6526.0 144.27 KB (147,728 bytes) 11/12/2000 8:47:22 AM Microsoft Corporationc:\winnt\system32\msdart.dll
sqloledb.dll 2000.080.0194 480.06 KB (491,584 bytes) 11/12/2000 8:47:29 AM Microsoft Corporationc:\program files\common files\system\ole db\sqloledb.dll
sqlftqry.dll2000.080.0194.00 108.07 KB (110,668 bytes) 11/12/2000 8:48:21 AM Microsoft Corporationc:\program files\microsoft sql server\mssql\binn\sqlftqry.dll
ssmslpcn.dll 2000.080.0194.00 28.06 KB (28,734 bytes) 11/12/2000 8:48:07 AM Microsoft Corporationc:\program files\microsoft sql server\mssql\binn\ssmslpcn.dll
security.dll5.00.2154.1 5.77 KB (5,904 bytes)12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\security.dll
ssnmpn70.dll 2000.080.0194.00 24.06 KB (24,638 bytes) 11/12/2000 8:48:07 AM Microsoft Corporationc:\program files\microsoft sql server\mssql\binn\ssnmpn70.dll
ssnetlib.dll2000.080.0194.00 84.06 KB (86,078 bytes) 11/12/2000 8:48:07 AM Microsoft Corporationc:\program files\microsoft sql server\mssql\binn\ssnetlib.dll
sqlevn70.rll 2000.080.0194.00 28.00 KB (28,672 bytes) 11/12/2000 8:48:08 AM Microsoft Corporationc:\program files\microsoft sql server\mssql\binn\resources\1033\sqlevn70.rll
msvcirt.dll 6.10.8637.0 76.05 KB (77,878 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\msvcirt.dll
sqlsort.dll 2000.080.0194.00 576.06 KB (589,885 bytes) 11/12/2000 8:48:07 AM Microsoft Corporationc:\program files\microsoft sql server\mssql\binn\sqlsort.dll
ums.dll 2000.080.0194.00 48.06 KB (49,210 bytes) 11/12/2000 8:48:07 AM Microsoft Corporationc:\program files\microsoft sql server\mssql\binn\ums.dll
opends60.dll 2000.080.0194.00 24.06 KB (24,639 bytes) 11/12/2000 8:48:07 AM Microsoft Corporationc:\program files\microsoft sql server\mssql\binn\opends60.dll
sqlservr_194_s2.exe 2000.080.0194.00 7.10 MB (7,442,493 bytes) 11/12/2000 9:02:16 AM Microsoft Corporationc:\program files\microsoft sql server\mssql\binn\sqlservr_194_s2.exe
cmd.exe 5.00.2144.1 230.77 KB (236,304 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\cmd.exe
sqlmangr.rll 2000.080.0194.00 96.00 KB (98,304 bytes) 11/12/2000 8:48:51 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\resources\1033\sqlmangr.rll
sqlsvc.rll 2000.080.0194.00 24.00 KB (24,576 bytes) 11/12/2000 8:48:48 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\resources\1033\sqlsvc.rll
odbcint.dll 3.520.6526.0 88.00 KB (90,112 bytes) 11/12/2000 8:47:24 AM Microsoft Corporationc:\winnt\system32\odbcint.dll
sqlresld.dll2000.080.0194.00 28.06 KB (28,738 bytes) 11/12/2000 8:48:48 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\sqlresld.dll
odbcbcp.dll2000.080.0194.00 28.07 KB (28,742 bytes) 11/12/2000 8:47:28 AM Microsoft Corporationc:\winnt\system32\odbcbcp.dll

sqlsvc.dll 2000.080.0194.00 92.06 KB (94,272 bytes) 11/12/2000 8:48:48 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\sqlsvc.dll
sqlunirl.dll 2000.080.0194.00 176.06 KB (180,290 bytes) 8/6/2000 2:51:56 AM Microsoft Corporationc:\winnt\system32\sqlunirl.dll
odbc32.dll 3.520.6526.0 216.27 KB (221,456 bytes) 11/12/2000 8:47:23 AM Microsoft Corporationc:\winnt\system32\odbc32.dll
w95scm.dll 2000.080.0194.00 48.06 KB (49,216 bytes) 11/12/2000 8:48:48 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\w95scm.dll
sqlmangr.exe 2000.080.0194.00 68.00 KB (69,632 bytes) 11/12/2000 8:48:51 AM Microsoft Corporationc:\program files\microsoft sql server\80\tools\binn\sqlmangr.exe
ibmmon.exe 1.11 28.50 KB (29,184 bytes) 11/12/2000 8:29:48 AM IBM Corporation c:\winnt\system32\ibmmon.exe
usp10.dll 1.0325.2180.1 307.77 KB (315,152 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\usp10.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
mshtmled.dll 5.00.2920.0000 230.27 KB (235,792 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\mshtmled.dll
webvw.dll 5.00.2920.0000 1.06 MB (1,115,408 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\webvw.dll
msls31.dll 3.10.337.0 145.27 KB (148,752 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\msls31.dll
msdbg.dll 6.00.8424 67.50 KB (69,120 bytes) 11/11/2000 6:14:32 AM Microsoft Corporationc:\winnt\system32\msdbg.dll
pdm.dll 6.00.8424 179.27 KB (183,574 bytes) 11/11/2000 6:14:32 AM Microsoft Corporationc:\winnt\system32\pdm.dll
mshtml.dll 5.00.2920.0000 2.25 MB (2,357,008 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\mshtml.dll
mlang.dll 5.00.2920.0000 510.77 KB (523,024 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\mlang.dll
urlmon.dll 5.00.2920.0000 426.77 KB (437,008 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\urlmon.dll
browselc.dll 5.00.2920.0000 34.50 KB (35,328 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\browselc.dll
shdoclc.dll 5.00.2920.0000 324.50 KB (332,288 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\shdoclc.dll
wininet.dll 5.00.2920.0000 456.77 KB (467,728 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\wininet.dll
faxshell.dll 5.00.2134.1 8.27 KB (8,464 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\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\system32\msacm32.dll
avifil32.dll 5.00.2134.1 76.27 KB (78,096 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\avifil32.dll
msvfw32.dll 5.00.2134.1 113.77 KB (116,496 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\msvfw32.dll
docprop2.dll 5.00.2178.1 297.77 KB (304,912 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\docprop2.dll
linkinfo.dll 5.00.2134.1 15.77 KB (16,144 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\linkinfo.dll
powrprof.dll 5.00.2920.0000 13.27 KB (13,584 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\powrprof.dll
batmeter.dll 5.00.2920.0000 20.27 KB (20,752 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\batmeter.dll
stobject.dll 5.00.2144.1 81.77 KB (83,728 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\stobject.dll
msi.dll 1.10.1029.0 1.71 MB (1,794,320 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\msi.dll
webcheck.dll 5.00.2920.0000 251.77 KB (257,808 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\webcheck.dll
ntshrui.dll 5.00.2134.1 46.77 KB (47,888 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\ntshrui.dll
mydocs.dll 5.00.2920.0000 55.77 KB (57,104 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\mydocs.dll
browseui.dll 5.00.2920.0000 793.27 KB (812,304 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\browseui.dll
shdocvw.dll 5.00.2920.0000 1.05 MB (1,104,144 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\shdocvw.dll
explorer.exe 5.00.2920.0000 232.77 KB (238,352 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\explorer.exe
netshell.dll 5.00.2176.1 456.77 KB (467,728 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\netshell.dll
netman.dll 5.00.2175.1 88.77 KB (90,896 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\netman.dll
es.dll 1999.9.3422.21 231.77 KB (237,328 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\es.dll
wshnetbs.dll 5.00.2134.1 7.77 KB (7,952 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\wshnetbs.dll
rapilib.dll 5.00.2167.1 25.27 KB (25,872 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\rapilib.dll
rsvpsp.dll 5.00.2167.1 74.77 KB (76,560 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\rsvpsp.dll
ntmarta.dll 5.00.2158.1 98.77 KB (101,136 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\ntmarta.dll
provthrd.dll 1.50.1085.0000 68.07 KB (69,708 bytes) 11/11/2000 11:17:06 AM Microsoft Corporation c:\winnt\system32\wbem\provthrd.dll
ntevt.dll 1.50.1085.0000 192.06 KB (196,669 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\wbem\ntevt.dll
perfos.dll 5.00.2155.1 21.27 KB (21,776 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\perfos.dll
wmi.dll 5.00.2191.1 6.27 KB (6,416 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\wmi.dll
psapi.dll 5.00.2134.1 28.27 KB (28,944 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\psapi.dll
framedyn.dll 1.50.1085.0000 164.05 KB (167,992 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\framedyn.dll
cimwin32.dll 1.50.1085.0000 1.03 MB (1,077,306 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\cimwin32.dll
wbemsvc.dll 1.50.1085.0000 140.07 KB (143,430 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\wbemsvc.dll
wbemess.dll 1.50.1085.0001 352.05 KB (360,503 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\wbemess.dll
fastprox.dll 1.50.1085.0001 144.08 KB (147,534 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\fastprox.dll
wbemcore.dll 1.50.1085.0001 632.05 KB (647,224 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\wbemcore.dll
wbemcomn.dll 1.50.1085.0001 684.05 KB (700,472 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\wbemcomn.dll
winmgmt.exe 1.50.1085.0001 188.05 KB (192,567 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\winmgmt.exe
simptcp.dll 5.00.2134.1 19.27 KB (19,728 bytes) 11/11/2000 6:11:20 AM Microsoft Corporationc:\winnt\system32\simptcp.dll
tcpsvcs.exe 5.00.2134.1 24.77 KB (25,360 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\tcpsvcs.exe
mtxoci.dll 1999.9.3421.3 109.27 KB (111,888 bytes) 11/11/2000 6:11:28 AM Microsoft Corporationc:\winnt\system32\mtxoci.dll

resutils.dll 5.00.2191.1 39.77 KB (40,720 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\resutils.dll
clusapi.dll 5.00.2179.1 50.27 KB (51,472 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\clusapi.dll
msvcp50.dll 5.00.7051 552.50 KB (565,760 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\msvcp50.dll
xolehlp.dll 1999.9.3421.3 17.27 KB (17,680 bytes) 11/11/2000 6:11:27 AM Microsoft Corporationc:\winnt\system32\xolehlp.dll
msdtclog.dll 1999.9.3421.3 89.77 KB (91,920 bytes) 11/11/2000 6:11:27 AM Microsoft Corporation c:\winnt\system32\msdtclog.dll
mtxclu.dll 1999.9.3421.3 50.27 KB (51,472 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\mtxclu.dll
msdtcprx.dll 1999.9.3422.10 619.27 KB (634,128 bytes) 11/11/2000 6:11:28 AM Microsoft Corporation c:\winnt\system32\msdtcprx.dll
txfaux.dll 1999.9.3422.24 341.27 KB (349,456 bytes) 11/11/2000 6:11:27 AM Microsoft Corporationc:\winnt\system32\txfaux.dll
msdtctm.dll 1999.9.3422.12 1.02 MB (1,070,864 bytes) 11/11/2000 6:11:27 AM Microsoft Corporation c:\winnt\system32\msdtctm.dll
msdtc.exe 1999.9.3421.3 6.77 KB (6,928 bytes)11/11/2000 6:11:27 AM Microsoft Corporationc:\winnt\system32\msdtc.exe
rpcss.dll 5.00.2181.1 229.27 KB (234,768 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\rpcss.dll
svchost.exe5.00.2134.1 7.77 KB (7,952 bytes)12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\svchost.exe
dssbase.dll 5.00.2150.1 140.77 KB (144,144 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\dssbase.dll
oakley.dll 5.00.2174.1 420.27 KB (430,352 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\oakley.dll
mfc42u.dll 6.00.8665.0 972.05 KB (995,384 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\mfc42u.dll
polagent.dll 5.00.2183.1 108.27 KB (110,864 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\polagent.dll
scecli.dll 5.00.2191.1 105.27 KB (107,792 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\scecli.dll
atl.dll 3.00.8449 57.56 KB (58,938 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\atl.dll
certcli.dll 5.00.2175.1 132.27 KB (135,440 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\certcli.dll
esent.dll 6.0.3939.6 1.07 MB (1,120,016 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\esent.dll
ntdsatq.dll 5.00.2181.1 31.27 KB (32,016 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\ntdsatq.dll
ntdsa.dll 5.00.2195.1 993.27 KB (1,017,104 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\ntdsa.dll
kdcsvc.dll 5.00.2181.1 133.77 KB (136,976 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\kdcsvc.dll
sfmapi.dll 5.00.2134.1 38.77 KB (39,696 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\sfmapi.dll
rassfm.dll 5.00.2168.1 21.27 KB (21,776 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\rassfm.dll
schannel.dll 5.00.2170.1 139.77 KB (143,120 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\schannel.dll
netlogon.dll 5.00.2182.1 347.77 KB (356,112 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\netlogon.dll
kerberos.dll 5.00.2181.1 196.77 KB (201,488 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\kerberos.dll
msprivs.dll5.00.2154.1 41.50 KB (42,496 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\msprivs.dll
samsrv.dll 5.00.2192.1 357.77 KB (366,352 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\samsrv.dll
lsasrv.dll 5.00.2184.1 487.77 KB (499,472 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\lsasrv.dll
lsass.exe 5.00.2184.1 32.77 KB (33,552 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\lsass.exe
ntlsapi.dll 5.00.2134.1 6.77 KB (6,928 bytes)12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\ntlsapi.dll
wmicore.dll 5.00.2178.1 70.77 KB (72,464 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\wmicore.dll
rasadhlp.dll 5.00.2168.1 7.27 KB (7,440 bytes)12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\rasadhlp.dll
winrnr.dll 5.00.2160.1 18.77 KB (19,216 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\winrnr.dll
dhcpcsvc.dll 5.00.2153.1 88.77 KB (90,896 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\dhcpcsvc.dll
tapi32.dll 5.00.2182.1 123.27 KB (126,224 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\tapi32.dll
rasman.dll 5.00.2188.1 54.77 KB (56,080 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\rasman.dll
rasapi32.dll 5.00.2188.1 189.77 KB (194,320 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\rasapi32.dll
rtutils.dll 5.00.2168.1 43.77 KB (44,816 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\rtutils.dll
adsldpc.dll5.00.2172.1 127.77 KB (130,832 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\adsldpc.dll
activeds.dll5.00.2172.1 172.77 KB (176,912 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\activeds.dll
mprapi.dll 5.00.2181.1 79.27 KB (81,168 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\mprapi.dll
icmp.dll 5.00.2134.1 7.27 KB (7,440 bytes)12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\icmp.dll
iphlpapi.dll5.00.2173.2 67.77 KB (69,392 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\iphlpapi.dll
rnr20.dll 5.00.2152.1 35.77 KB (36,624 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\rnr20.dll
wshtcpip.dll 5.00.2134.1 17.27 KB (17,680 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\wshtcpip.dll
msafd.dll 5.00.2153.1 54.27 KB (55,568 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\msafd.dll
mswsock.dll 5.00.2152.1 62.27 KB (63,760 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\mswsock.dll
msgsvc.dll 5.00.2181.1 33.77 KB (34,576 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\msgsvc.dll
browser.dll5.00.2142.1 48.27 KB (49,424 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\browser.dll
alrsvc.dll 5.00.2134.1 17.77 KB (18,192 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\alrsvc.dll
psbase.dll 5.00.2146.1 111.77 KB (114,448 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\psbase.dll
seclogon.dll 5.00.2135.1 15.77 KB (16,144 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\seclogon.dll
cryptsvc.dll 5.00.2181.1 61.77 KB (63,248 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\cryptsvc.dll
cryptdll.dll5.00.2135.1 41.27 KB (42,256 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\cryptdll.dll
wkssvc.dll 5.00.2181.1 95.27 KB (97,552 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\wkssvc.dll
srvsvc.dll 5.00.2178.1 79.27 KB (81,168 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\srvsvc.dll
cfgmgr32.dll 5.00.2134.1 16.77 KB (17,168 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\cfgmgr32.dll
dmserver.dll 2191.1.296.2 11.77 KB (12,048 bytes) 12/7/1999 6:00:00 AM VERITAS Software Corp. c:\winnt\system32\dmserver.dll
winsta.dll 5.00.2134.1 36.27 KB (37,136 bytes) 12/7/1999 6:00:00 AM Microsoft Corporationc:\winnt\system32\winsta.dll

eventlog.dll 5.00.2178.1 43.77 KB (44,816 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\eventlog.dll
ntdsapi.dll 5.00.2160.1 56.27 KB (57,616 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\ntdsapi.dll
scesrv.dll 5.00.2188.1 225.77 KB (231,184 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\scesrv.dll
umpnpmgr.dll 5.00.2182.1 86.27 KB (88,336 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\umpnpmgr.dll
services.exe 5.00.2134.1 86.77 KB (88,848 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\services.exe
msv1_0.dll 5.00.2164.1 94.77 KB (97,040 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\msv1_0.dll
clbcatq.dll 1999.9.3422.14 479.27 KB (490,768 bytes) 11/11/2000 6:11:20 AM Microsoft Corporation c:\winnt\system32\clbcatq.dll
oleaut32.dll 2.40.4512 600.27 KB (614,672 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\oleaut32.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
comdlg32.dll 5.00.2920.0000 236.77 KB (242,448 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\comdlg32.dll
netui2.dll 5.00.2134.1 280.27 KB (286,992 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\netui2.dll
mprui.dll 5.00.2134.1 54.77 KB (56,080 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\mprui.dll
netui1.dll 5.00.2134.1 210.27 KB (215,312 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\netui1.dll
netui0.dll 5.00.2134.1 70.27 KB (71,952 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\netui0.dll
ntlanman.dll 5.00.2157.1 35.27 KB (36,112 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\ntlanman.dll
mpr.dll 5.00.2146.1 53.27 KB (54,544 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\mpr.dll
cscui.dll 5.00.2172.1 227.27 KB (232,720 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\cscui.dll
winspool.drv 5.00.2167.1 109.77 KB (112,400 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\winspool.drv
winscard.dll 5.00.2134.1 77.27 KB (79,120 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\winscard.dll
wlnotify.dll 5.00.2164.1 53.27 KB (54,544 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\wlnotify.dll
cscdll.dll 5.00.2189.1 98.27 KB (100,624 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\cscdll.dll
lz32.dll 5.00.2134.1 9.77 KB (10,000 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\lz32.dll
version.dll 5.00.2134.1 15.77 KB (16,144 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\version.dll
rsabase.dll 5.00.2150.1 128.77 KB (131,856 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\rsabase.dll
mscat32.dll 5.131.2134.1 7.77 KB (7,952 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\mscat32.dll
ole32.dll 5.00.2181.1 966.27 KB (989,456 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\ole32.dll
imagehlp.dll 5.00.2195.1 125.27 KB (128,272 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\imagehlp.dll
msasn1.dll 5.00.2134.1 51.27 KB (52,496 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\msasn1.dll
crypt32.dll 5.131.2173.1 465.77 KB (476,944 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\crypt32.dll
wintrust.dll 5.131.2143.1 162.27 KB (166,160 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\wintrust.dll
setupapi.dll 5.00.2183.1 554.27 KB (567,568 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\setupapi.dll
winmm.dll 5.00.2161.1 184.77 KB (189,200 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\winmm.dll
comctl32.dll 5.81 540.27 KB (553,232 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\comctl32.dll
shlwapi.dll 5.00.2920.0000 282.77 KB (289,552 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\shlwapi.dll
shell32.dll 5.00.2920.0000 2.24 MB (2,352,400 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\shell32.dll
msgina.dll 5.00.2191.1 309.77 KB (317,200 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\msgina.dll
wsock32.dll 5.00.2152.1 21.27 KB (21,776 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\wsock32.dll
dnsapi.dll 5.00.2181.1 129.77 KB (132,880 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\dnsapi.dll
wldap32.dll 5.00.2168.1 155.77 KB (159,504 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\wldap32.dll
ws2help.dll 5.00.2134.1 17.77 KB (18,192 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\ws2help.dll
ws2_32.dll 5.00.2134.1 69.77 KB (71,440 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\ws2_32.dll
samlib.dll 5.00.2160.1 46.27 KB (47,376 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\samlib.dll
netrap.dll 5.00.2134.1 11.27 KB (11,536 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\netrap.dll
netapi32.dll 5.00.2194.1 302.77 KB (310,032 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\netapi32.dll
profmap.dll 5.00.2181.1 29.27 KB (29,968 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\profmap.dll
secur32.dll 5.00.2154.1 46.77 KB (47,888 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\secur32.dll
sfc.dll 5.00.2164.1 84.27 KB (86,288 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\sfc.dll
nddeapi.dll 5.00.2137.1 15.27 KB (15,632 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\nddeapi.dll
userenv.dll 5.00.2185.1 361.27 KB (369,936 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\userenv.dll
user32.dll 5.00.2180.1 393.27 KB (402,704 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\user32.dll
gdi32.dll 5.00.2180.1 228.77 KB (234,256 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\gdi32.dll
rpcrt4.dll 5.00.2193.1 434.27 KB (444,688 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\rpcrt4.dll
advapi32.dll 5.00.2191.1 349.27 KB (357,648 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\advapi32.dll
kernel32.dll 5.00.2191.1 715.27 KB (732,432 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\kernel32.dll
msvcrt.dll 6.10.8637.0 288.09 KB (295,000 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\msvcrt.dll
winlogon.exe 5.00.2182.1 173.27 KB (177,424 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\winlogon.exe
sfcfiles.dll 5.00.2195.1 973.27 KB (996,624 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\sfcfiles.dll
ntdll.dll 5.00.2163.1 469.77 KB (481,040 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\ntdll.dll
smss.exe 5.00.2170.1 44.27 KB (45,328 bytes) 12/7/1999 6:00:00 AM Microsoft Corporation c:\winnt\system32\smss.exe

[Services]

Display Name Name State Start Mode Service Type Path Error Control Start Name Tag ID

Alerter Alerter 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 Stopped Manual Own Process
c:\winnt\system32\dfssvc.exe Normal LocalSystem 0
DHCP Client Dhcp Stopped Manual 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 Stopped Manual 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
Mylex Global Array Manager Server gamscm Running Manual
Own Process c:\winnt\system32\gamserv\gamscm.exe Normal
LocalSystem 0
Internet Authentication Service IAS Stopped Manual Share
Process c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem
0
IIS Admin Service IISADMIN Stopped Manual Share
Process c:\winnt\system32\inetsrv\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 c:\winnt\system32\lsass.exe Normal LocalSystem 0
Server lanmanserver Running Manual 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 Stopped Manual
Own Process c:\winnt\system32\llssrv.exe Normal LocalSystem
0
TCP/IP NetBIOS Helper Service LmHosts Stopped Manual 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 mnmsrvc Stopped Manual
Own Process c:\winnt\system32\mnmsrvc.exe Normal LocalSystem
0
Distributed Transaction Coordinator MSDTC Running Auto
Own Process c:\winnt\system32\msdtc.exe Normal LocalSystem
0
Windows Installer MSIServer Stopped Manual Share Process
c:\winnt\system32\msiexec.exe /v Normal LocalSystem 0
Microsoft Search MSSEARCH Stopped Manual Share
Process "c:\program files\common files\system\mssearch\bin\mssearch.exe"
Normal LocalSystem 0
MSSQLSERVER MSSQLSERVER Stopped Manual Own Process
c:\progra~1\micros~3\mssql\binn\sqlservr.exe Normal LocalSystem 0
MSSQLServerADHelper MSSQLServerADHelper Stopped
Manual Own Process c:\program files\microsoft sql
server\80\tools\binn\sqladhlp.exe Normal LocalSystem 0
Network DDE NetDDE Stopped Manual Share Process
c:\winnt\system32\netdde.exe Normal LocalSystem 0
Network DDE DSDM NetDDEdsdm Stopped Manual Share
Process c:\winnt\system32\netdde.exe Normal LocalSystem 0
Net Logon Netlogon Stopped Manual Share Process
c:\winnt\system32\lsass.exe Normal LocalSystem 0
Network Connections Netman Running Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
File Replication NtFrs Stopped Manual Own Process
c:\winnt\system32\ntfrs.exe Ignore LocalSystem 0
NT LM Security Support Provider NtLmSsp Stopped Manual Share
Process c:\winnt\system32\lsass.exe Normal LocalSystem 0
Removable Storage NtmsSvc Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Plug and Play PlugPlay Running Auto Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
IPSEC Policy Agent PolicyAgent Running Auto Share
Process c:\winnt\system32\lsass.exe Normal LocalSystem 0
Protected Storage ProtectedStorage Running Auto Share
Process c:\winnt\system32\services.exe Normal LocalSystem 0
Remote Access Auto Connection Manager RasAuto Stopped Manual
Share Process c:\winnt\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Remote Access Connection Manager RasMan Stopped Manual
Share Process c:\winnt\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Routing and Remote Access RemoteAccess Stopped Disabled
Share Process c:\winnt\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Remote Registry Service RemoteRegistry Stopped Manual
Own Process c:\winnt\system32\regsvc.exe Normal LocalSystem
0
Remote Procedure Call (RPC) Locator RpcLocator Stopped
Manual Own Process c:\winnt\system32\locator.exe Normal
LocalSystem 0
Remote Procedure Call (RPC) RpcSs Running Manual Share
Process c:\winnt\system32\svchost -k rpcss Normal LocalSystem 0
QoS Admission Control (RSVP) RSVP Running Auto Own Process
c:\winnt\system32\rsvp.exe -s Normal LocalSystem 0
Security Accounts Manager SamSs Running Auto Share
Process c:\winnt\system32\lsass.exe Normal LocalSystem 0
Smart Card Helper SCardDrv Stopped Manual Share Process
c:\winnt\system32\scardsvr.exe Ignore LocalSystem 0
Smart Card SCardSvr Stopped Manual Share Process
c:\winnt\system32\scardsvr.exe Ignore LocalSystem 0
Task Scheduler Schedule Stopped Manual Share Process
c:\winnt\system32\mstask.exe Normal LocalSystem 0
RunAs Service seclogon Running Auto Share Process
c:\winnt\system32\services.exe Ignore LocalSystem 0
System Event Notification SENS Stopped Manual Share
Process c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem
0
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 Share
Process c:\winnt\system32\tcpsvcs.exe Normal LocalSystem 0
Print Spooler Spooler Stopped Manual Own Process
c:\winnt\system32\spoolsv.exe Normal LocalSystem 0
SQLSERVERAGENT SQLSERVERAGENT Stopped Manual Own Process
c:\progra~1\micros~3\mssql\binn\sqlagent.exe Normal LocalSystem 0
Performance Logs and Alerts SysmonLog Stopped Manual
Own Process c:\winnt\system32\smlogsvc.exe Normal LocalSystem
0
Telephony TapiSrv Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k tapisrv Normal LocalSystem 0
Terminal Services TermService Stopped Disabled Own Process
c:\winnt\system32\termsrv.exe Normal LocalSystem 0
Telnet TlntSvr Stopped Manual Own Process
c:\winnt\system32\tlntsvr.exe Normal LocalSystem 0
Distributed Link Tracking Server TrkSvr Stopped Manual Share
Process c:\winnt\system32\services.exe Normal LocalSystem 0

Distributed Link Tracking Client TrkWks Stopped Manual Share Process c:\winnt\system32\services.exe Normal LocalSystem 0
Uninterruptible Power Supply UPS Stopped Manual Own Process c:\winnt\system32\ups.exe Normal LocalSystem 0
Utility Manager UtilMan Stopped Manual Own Process c:\winnt\system32\utilman.exe Normal LocalSystem 0
Windows Time W32Time Stopped Manual Share Process c:\winnt\system32\services.exe Normal LocalSystem 0
World Wide Web Publication Service W3SVC Stopped Manual Share Process c:\winnt\system32\inetsrv\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
Microsoft SQL Server - Switch All Users:Microsoft SQL Server - Switch All Users
Startup All Users:Startup All Users
Accessories IBMSERV3\Administrator:Accessories IBMSERV3\Administrator
Accessories\Accessibility IBMSERV3\Administrator:Accessories\Accessibility IBMSERV3\Administrator
Accessories\Entertainment IBMSERV3\Administrator:Accessories\Entertainment IBMSERV3\Administrator
Accessories\System Tools IBMSERV3\Administrator:Accessories\System Tools IBMSERV3\Administrator
Startup IBMSERV3\Administrator:Startup IBMSERV3\Administrator

[Startup Programs]

Program Command User NameLocation
Service Manager c:\progra~1\micros~3\80\tools\binn\sqlmangr.exe /n All Users Common Startup
Ibmmon.exe ibmmon.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:\PROGRA~1\WINDOW~1\ACCESS~1\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 5.00.2920.0000
Build 52920
Product ID 51879-270-9401932-05426
Application Path C:\Program Files\Internet Explorer
Language English (United States)
Active Printer Not Available

Cipher Strength 56-bit
Content Advisor Disabled
IEAK Install No

[File Versions]

File Version Size Date Path Company
advapi32.dll 5.0.2191.1 349 KB 12/7/1999 6:00:00 AM C:\WINNT\system32 Microsoft Corporation
advpack.dll 5.0.2920.0 87 KB 12/7/1999 6:00:00 AM C:\WINNT\system32 Microsoft Corporation
browselc.dll 5.0.2920.0 35 KB 12/7/1999 6:00:00 AM C:\WINNT\system32 Microsoft Corporation
browseui.dll 5.0.2920.0 793 KB 12/7/1999 6:00:00 AM C:\WINNT\system32 Microsoft Corporation
ckcnv.exe 5.0.2189.1 9 KB 12/7/1999 6:00:00 AM C:\WINNT\system32 Microsoft Corporation
comctl32.dll 5.81.2920.0 540 KB 12/7/1999 6:00:00 AM C:\WINNT\system32 Microsoft Corporation
crypt32.dll 5.131.2173.1 466 KB 12/7/1999 6:00:00 AM C:\WINNT\system32 Microsoft Corporation
enhsig.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 5.0.2920.0 57 KB 12/7/1999 6:00:00 AM C:\WINNT\system32 Microsoft Corporation
iexplore.exe 5.0.2920.0 59 KB 12/7/1999 6:00:00 AM C:\Program Files\Internet Explorer Microsoft Corporation
imagehlp.dll 5.0.2195.1 125 KB 12/7/1999 6:00:00 AM C:\WINNT\system32 Microsoft Corporation
imghelp.dll <File Missing> Not Available Not Available Not Available Not Available
inseng.dll 5.0.2920.0 72 KB 12/7/1999 6:00:00 AM C:\WINNT\system32 Microsoft Corporation
jobexec.dll 5.0.0.1 47 KB 12/7/1999 6:00:00 AM C:\WINNT\system32 Microsoft Corporation
jscript.dll 5.1.0.4615 476 KB 12/7/1999 6:00:00 AM C:\WINNT\system32 Microsoft Corporation
jsproxy.dll 5.0.2920.0 13 KB 12/7/1999 6:00:00 AM C:\WINNT\system32 Microsoft Corporation
msaahtml.dll <File Missing> Not Available Not Available Not Available Not Available
mshtml.dll 5.0.2920.0 2302 KB 12/7/1999 6:00:00 AM C:\WINNT\system32 Microsoft Corporation

msjava.dll 5.0.3234.0 918 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
msoss.dll <File Missing> Not Available Not Available
Not Available Not Available
msxml.dll 5.0.2920.0 521 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
occache.dll 5.0.2920.0 86 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
ole32.dll 5.0.2181.1 966 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
oleaut32.dll 2.40.4512.1 600 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
olepro32.dll 5.0.4512.1 160 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
rsabase.dll 5.0.2150.1 129 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
rsaenh.dll <File Missing> Not Available Not Available
Not Available Not Available
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 5.0.2170.0 140 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
shdoc401.dll <File Missing> Not Available Not
Available Not Available Not Available
shdocvw.dll 5.0.2920.0 1078 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
shell32.dll 5.0.2920.0 2297 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
shlwapi.dll 5.0.2920.0 283 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
url.dll 5.0.2920.0 82 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
urlmon.dll 5.0.2920.0 427 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
vbscript.dll 5.1.0.4615 428 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
webcheck.dll 5.0.2920.0 252 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
win.com 5.0.2134.1 24 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
wininet.dll 5.0.2920.0 457 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
winsock.dll 3.10.0.103 3 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
wintrust.dll 5.131.2143.1 162 KB 12/7/1999 6:00:00 AM
C:\WINNT\system32 Microsoft Corporation
wsock.vxd <File Missing> Not Available Not Available
Not Available Not Available
wsock32.dll 5.0.2152.1 21 KB 12/7/1999 6:00:00 AM
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 Not Available
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 4698 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 11/12/2000 to 10/19/2100
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

Microsoft Windows 2000 Advanced Server Services

```
Name    Description    Status         Startup Type
Log On As
Alerter Notifies selected users and computers of
administrative alerts. Started Automatic
LocalSystem
Application Management Provides software installation
services such as Assign, Publish, and Remove.
Started Manual  LocalSystem
ClipBook       Supports ClipBook Viewer, which allows
pages to be seen by remote ClipBooks.        Manual
LocalSystem
```

```
COM+ Event System      Provides automatic distribution
of events to subscribing COM components.      Started
Manual  LocalSystem
Computer Browser       Maintains an up-to-date list of
computers on your network and supplies the list to
programs that request it.      Started Automatic
LocalSystem
DHCP Client    Manages network configuration by
registering and updating IP addresses and DNS names.
Started Automatic      LocalSystem
Distributed File System        Manages logical volumes
distributed across a local or wide area network.
Started Automatic      LocalSystem
Distributed Link Tracking Client       Sends
notifications of files moving between NTFS volumes in
a network domain.      Started Automatic
LocalSystem
Distributed Link Tracking Server       Stores
information so that files moved between volumes can
be tracked for each volume in the domain.
Manual  LocalSystem
Distributed Transaction Coordinator    Coordinates
transactions that are distributed across two or more
databases, message queues, file systems, or other
transaction protected resource managers.      Started
Automatic      LocalSystem
DNS Client     Resolves and caches Domain Name System
(DNS) names.           Manual  LocalSystem
Event Log      Logs event messages issued by programs
and Windows.  Event Log reports contain information
that can be useful in diagnosing problems.  Reports
are viewed in Event Viewer.    Started Automatic
LocalSystem
Fax Service    Helps you send and receive faxes
Manual  LocalSystem
File Replication       Maintains file synchronization
of file directory contents among multiple servers.
Manual  LocalSystem
FTP Publishing Service Provides FTP connectivity and
administration through the Internet Information
Services snap-in.              Manual  LocalSystem
IIS Admin Service      Allows administration of Web
and FTP services through the Internet Information
Services snap-in.              Manual  LocalSystem
Indexing Service       Indexes contents and properties
of files on local and remote computers; provides
rapid access to files through flexible querying
language.              Manual  LocalSystem
Internet Authentication Service        Enables
authentication, authorization and accounting of
dial-up and VPN users. IAS supports the RADIUS
protocol.              Manual  LocalSystem
Internet Connection Sharing    Provides network
address translation, addressing, and name resolution
services for all computers on your home network
through a dial-up connection.          Manual
LocalSystem
Intersite Messaging    Allows sending and receiving
messages between Windows Advanced Server sites.
Disabled       LocalSystem
IPSEC Policy Agent     Manages IP security policy and
starts the ISAKMP/Oakley (IKE) and the IP security
driver. Started Automatic      LocalSystem
Kerberos Key Distribution Center       Generates
session keys and grants service tickets for mutual
client/server authentication.          Disabled
LocalSystem
License Logging Service                        Manual
LocalSystem
Logical Disk Manager   Logical Disk Manager Watchdog
Service Started Automatic      LocalSystem
Logical Disk Manager Administrative Service
Administrative service for disk management requests
Manual  LocalSystem
Messenger      Sends and receives messages
transmitted by administrators or by the Alerter
service.       Started Automatic      LocalSystem
Microsoft Search       Creates full-text indexes on
content and properties of structured and
semi-structured data to allow fast linguistic
searches on this data  Started Automatic
LocalSystem
MSSQLSERVER                    Manual  LocalSystem
MSSQLServerADHelper                    Manual
LocalSystem
Mylex Global Array Manager Server
Manual  LocalSystem
Net Logon      Supports pass-through authentication
of account logon events for computers in a domain.
Manual  LocalSystem
NetMeeting Remote Desktop Sharing      Allows
authorized people to remotely access your Windows
desktop using NetMeeting.              Manual
LocalSystem
Network Connections    Manages objects in the Network
and Dial-Up Connections folder, in which you can view
both local area network and remote connections.
Started Manual  LocalSystem
Network DDE    Provides network transport and
security for dynamic data exchange (DDE).
Manual  LocalSystem
Network DDE DSDM       Manages shared dynamic data
exchange and is used by Network DDE            Manual
LocalSystem
NT LM Security Support Provider        Provides
security to remote procedure call (RPC) programs that
use transports other than named pipes.       Started
Manual  LocalSystem
Performance Logs and Alerts    Configures performance
logs and alerts.               Manual  LocalSystem
Plug and Play  Manages device installation and
configuration and notifies programs of device
changes.       Started Automatic      LocalSystem
Print Spooler  Loads files to memory for later
printing.              Manual  LocalSystem
Protected Storage      Provides protected storage for
sensitive data, such as private keys, to prevent
access by unauthorized services, processes, or users.
Started Automatic      LocalSystem
QoS Admission Control (RSVP)  Provides network
signaling and local traffic control setup
functionality for QoS-aware programs and control
applets.               Automatic      LocalSystem
Remote Access Auto Connection Manager Creates a
connection to a remote network whenever a program
references a remote DNS or NetBIOS name or address.
Manual  LocalSystem
Remote Access Connection Manager       Creates a
network connection.            Manual  LocalSystem
Remote Procedure Call (RPC)    Provides the endpoint
mapper and other miscellaneous RPC services. Started
Automatic      LocalSystem
Remote Procedure Call (RPC) Locator    Manages the RPC
name service database.         Manual  LocalSystem
Remote Registry Service        Allows remote registry
manipulation.  Started Automatic       LocalSystem
Removable Storage      Manages removable media,
drives, and libraries. Started Automatic
LocalSystem
Routing and Remote Access      Offers routing services
to businesses in local area and wide area network
environments.          Disabled       LocalSystem
RunAs Service  Enables starting processes under
alternate credentials  Started Automatic
LocalSystem
Security Accounts Manager      Stores security
information for local user accounts.  Started
Automatic      LocalSystem
```

```
Server  Provides RPC support and file, print, and
named pipe sharing.    Started Automatic
LocalSystem
Simple TCP/IP Services Supports the following TCP/IP
services: Character Generator, Daytime, Discard,
Echo, and Quote of the Day.          Manual
LocalSystem
Smart Card     Manages and controls access to a smart
card inserted into a smart card reader attached to
the computer.          Manual  LocalSystem
Smart Card Helper      Provides support for legacy
smart card readers attached to the computer.
Manual  LocalSystem
SNMP Service   Includes agents that monitor the
activity in network devices and report to the network
console workstation.          Manual  LocalSystem
SNMP Trap Service      Receives trap messages
generated by local or remote SNMP agents and forwards
the messages to SNMP management programs running on
this computer.         Manual  LocalSystem
SQLSERVERAGENT                 Manual  LocalSystem
System Event Notification      Tracks system events
such as Windows logon, network, and power events.
Notifies COM+ Event System subscribers of these
events. Started Automatic      LocalSystem
Task Scheduler Enables a program to run at a
designated time.       Started Automatic
LocalSystem
TCP/IP NetBIOS Helper Service Enables support for
NetBIOS over TCP/IP (NetBT) service and NetBIOS name
resolution.    Started Automatic      LocalSystem
Telephony      Provides Telephony API (TAPI) support
for programs that control telephony devices and IP
based voice connections on the local computer and,
through the LAN, on servers that are also running the
service.       Started Manual  LocalSystem
Telnet  Allows a remote user to log on to the system
and run console programs using the command line.
Manual  LocalSystem
Terminal Services      Provides a multisession
environment that allows client devices to access a
virtual Windows 2000 Professional desktop session and
Windows-based programs running on the server.
Disabled       LocalSystem
Uninterruptible Power Supply  Manages an
uninterruptible power supply (UPS) connected to the
computer.              Manual  LocalSystem
Utility Manager        Starts and configures
accessibility tools from one window          Manual
LocalSystem
Windows Installer                      Manual
LocalSystem
Windows Management Instrumentation    Provides system
management information.       Started Automatic
LocalSystem
Windows Management Instrumentation Driver Extensions
Provides systems management information to and from
drivers.       Started Manual  LocalSystem
Windows Time   Sets the computer clock.
Manual  LocalSystem
Workstation    Provides network connections and
communications.        Started Automatic
LocalSystem
World Wide Web Publishing Service     Provides Web
connectivity and administration through the Internet
Information Services snap-in.        Manual
LocalSystem
```

Microsoft Windows 2000 Advanced Server Registry Parameters

The following registry parameters have been modified:

```
Key Name:

SYSTEM\CurrentControlSet\Services\dac2w2k\Parameters\
Device0
Class Name:        <NO CLASS>
Last Write Time:   10/13/2000 - 9:06 AM
Value 0
  Name:            DriverParameter
  Type:            REG_SZ
  Data:            ConfigureSIR=24
Key Name:
SYSTEM\CurrentControlSet\Control\Session Manager\I/O
System
Class Name:        <NO CLASS>
Last Write Time:   10/8/2000 - 1:58 PM
Value 0
  Name:            CountOperations
  Type:            REG_DWORD
  Data:            0

Value 1
  Name:            LargeIrpStackLocations
  Type:            REG_DWORD
  Data:            0x8

Key Name:
SYSTEM\CurrentControlSet\Control\Session
Manager\Memory Management
Class Name:        <NO CLASS>
Last Write Time:   10/8/2000 - 1:58 PM
Value 0
  Name:            ClearPageFileAtShutdown
  Type:            REG_DWORD
  Data:            0

Value 1
  Name:            DisablePagingExecutive
  Type:            REG_DWORD
  Data:            0

Value 2
  Name:            IoPageLockLimit
  Type:            REG_DWORD
  Data:            0

Value 3
  Name:            LargeSystemCache
  Type:            REG_DWORD
  Data:            0

Value 4
  Name:            NonPagedPoolQuota
  Type:            REG_DWORD
  Data:            0

Value 5
  Name:            NonPagedPoolSize
  Type:            REG_DWORD
  Data:            0

Value 6
  Name:            PagedPoolQuota
  Type:            REG_DWORD
  Data:            0

Value 7
  Name:            PagedPoolSize
  Type:            REG_DWORD
  Data:            0

Value 8
  Name:            PagingFiles
  Type:            REG_MULTI_SZ
  Data:            C:\pagefile.sys 2046 4092
```

```
Value 9
  Name:             PhysicalAddressExtension
  Type:             REG_DWORD
  Data:             0x1

Value 10
  Name:             SecondLevelDataCache
  Type:             REG_DWORD
  Data:             0

Value 11
  Name:             SystemPages
  Type:             REG_DWORD
  Data:             0
```

Disk Controller Configuration Parameters

Mylex eXtremeRAID Adapter 0

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;

PhysicalDevice42 = Channel=3, Target=0, Size=17340mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice43 = Channel=3, Target=1, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice44 = Channel=3, Target=2, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice45 = Channel=3, Target=3, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice46 = Channel=3, Target=4, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice47 = Channel=3, Target=5, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice48 = Channel=3, Target=6, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice49 = Channel=3, Target=8, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice50 = Channel=3, Target=9, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice51 = Channel=3, Target=10, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice52 = Channel=3, Target=11, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice53 = Channel=3, Target=12, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice54 = Channel=3, Target=13, Size=17340mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice55 = Channel=3, 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);
        IntermediateDevice3 = StripeSize=64kb, Raid=0, WriteThrough=1,
Size=242704mb,
                (PhysicalDevice42, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice43, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice44, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice45, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice46, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice47, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice48, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice49, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice50, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice51, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice52, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice53, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice54, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice55, StartAddress=0mb, Size=17336mb);
        LogicalDevice0 = StripeSize=64kb, Raid=12, WriteThrough=1,
Size=970816mb, BIOSGeometry=8GB,
                (IntermediateDevice0, StartAddress=0mb,
Size=242704mb),
                (IntermediateDevice1, StartAddress=0mb,
Size=242704mb),
                (IntermediateDevice2, StartAddress=0mb,
Size=242704mb),
                (IntermediateDevice3, StartAddress=0mb,
Size=242704mb);
 EndGroup
 BeginControllerParameter
        ControllerName = eXtremeRAID 2000;
        ControllerType = 28;
        FirmwareVersion = 5.60;
        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 = 1;
        DisableBios = 0;
        EnableCDROMBoot = 0;
        EnableStorageWorks = 0;
        EnableSAFTE = 0;
        EnableSES = 0;
        EnableARM = 0;
        EnableOFM = 1;
        OEMCode = 0;
        StartupOption = 0;
 EndControllerParameter
End
```

Mylex eXtremeRAID Adapter 1

```
Begin
 BeginGroup
```

PhysicalDevice0 = Channel=0, Target=0, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice1 = Channel=0, Target=1, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice2 = Channel=0, Target=2, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice3 = Channel=0, Target=3, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice4 = Channel=0, Target=4, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice5 = Channel=0, Target=5, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice6 = Channel=0, Target=6, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice7 = Channel=0, Target=8, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice8 = Channel=0, Target=9, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice9 = Channel=0, Target=10, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice10 = Channel=0, Target=11, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice11 = Channel=0, Target=12, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice12 = Channel=0, Target=13, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice13 = Channel=0, Target=14, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice14 = Channel=1, Target=0, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice15 = Channel=1, Target=1, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice16 = Channel=1, Target=2, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice17 = Channel=1, Target=3, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice18 = Channel=1, Target=4, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice19 = Channel=1, Target=5, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice20 = Channel=1, Target=6, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice21 = Channel=1, Target=8, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice22 = Channel=1, Target=9, Size=8662mb, State=Online,
TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice23 = Channel=1, Target=10, Size=8662mb, State=Online,

```
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice24 = Channel=1, Target=11, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice25 = Channel=1, Target=12, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice26 = Channel=1, Target=13, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice27 = Channel=1, Target=14, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice28 = Channel=2, Target=0, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice29 = Channel=2, Target=1, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice30 = Channel=2, Target=2, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice31 = Channel=2, Target=3, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice32 = Channel=2, Target=4, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice33 = Channel=2, Target=5, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice34 = Channel=2, Target=6, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice35 = Channel=2, Target=8, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice36 = Channel=2, Target=9, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice37 = Channel=2, Target=10, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice38 = Channel=2, Target=11, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice39 = Channel=2, Target=12, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice40 = Channel=2, Target=13, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice41 = Channel=2, Target=14, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice42 = Channel=3, Target=0, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice43 = Channel=3, Target=1, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice44 = Channel=3, Target=2, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice45 = Channel=3, Target=3, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        PhysicalDevice46 = Channel=3, Target=4, Size=8662mb,
State=Online,

                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
```

PhysicalDevice47 = Channel=3, Target=5, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice48 = Channel=3, Target=6, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice49 = Channel=3, Target=8, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice50 = Channel=3, Target=9, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice51 = Channel=3, Target=10, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice52 = Channel=3, Target=11, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice53 = Channel=3, Target=12, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice54 = Channel=3, Target=13, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice55 = Channel=3, Target=14, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

IntermediateDevice0 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=121184mb,

(PhysicalDevice0, StartAddress=0mb, Size=8656mb),

(PhysicalDevice1, StartAddress=0mb, Size=8656mb),

(PhysicalDevice2, StartAddress=0mb, Size=8656mb),

(PhysicalDevice3, StartAddress=0mb, Size=8656mb),

(PhysicalDevice4, StartAddress=0mb, Size=8656mb),

(PhysicalDevice5, StartAddress=0mb, Size=8656mb),

(PhysicalDevice6, StartAddress=0mb, Size=8656mb),

(PhysicalDevice7, StartAddress=0mb, Size=8656mb),

(PhysicalDevice8, StartAddress=0mb, Size=8656mb),

(PhysicalDevice9, StartAddress=0mb, Size=8656mb),

(PhysicalDevice10, StartAddress=0mb, Size=8656mb),

(PhysicalDevice11, StartAddress=0mb, Size=8656mb),

(PhysicalDevice12, StartAddress=0mb, Size=8656mb),

(PhysicalDevice13, StartAddress=0mb, Size=8656mb);

IntermediateDevice1 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=121184mb,

(PhysicalDevice14, StartAddress=0mb, Size=8656mb),

(PhysicalDevice15, StartAddress=0mb, Size=8656mb),

(PhysicalDevice16, StartAddress=0mb, Size=8656mb),

(PhysicalDevice17, StartAddress=0mb, Size=8656mb),

(PhysicalDevice18, StartAddress=0mb, Size=8656mb),

(PhysicalDevice19, StartAddress=0mb, Size=8656mb),

(PhysicalDevice20, StartAddress=0mb, Size=8656mb),

(PhysicalDevice21, StartAddress=0mb, Size=8656mb),

(PhysicalDevice22, StartAddress=0mb, Size=8656mb),

(PhysicalDevice23, StartAddress=0mb, Size=8656mb),

(PhysicalDevice24, StartAddress=0mb, Size=8656mb),

(PhysicalDevice25, StartAddress=0mb, Size=8656mb),

(PhysicalDevice26, StartAddress=0mb, Size=8656mb),

(PhysicalDevice27, StartAddress=0mb, Size=8656mb);

IntermediateDevice2 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=121184mb,

(PhysicalDevice28, StartAddress=0mb, Size=8656mb),

(PhysicalDevice29, StartAddress=0mb, Size=8656mb),

(PhysicalDevice30, StartAddress=0mb, Size=8656mb),

(PhysicalDevice31, StartAddress=0mb, Size=8656mb),

(PhysicalDevice32, StartAddress=0mb, Size=8656mb),

(PhysicalDevice33, StartAddress=0mb, Size=8656mb),

(PhysicalDevice34, StartAddress=0mb, Size=8656mb),

(PhysicalDevice35, StartAddress=0mb, Size=8656mb),

(PhysicalDevice36, StartAddress=0mb, Size=8656mb),

(PhysicalDevice37, StartAddress=0mb, Size=8656mb),

(PhysicalDevice38, StartAddress=0mb, Size=8656mb),

(PhysicalDevice39, StartAddress=0mb, Size=8656mb),

(PhysicalDevice40, StartAddress=0mb, Size=8656mb),

(PhysicalDevice41, StartAddress=0mb, Size=8656mb);

IntermediateDevice3 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=121184mb,

(PhysicalDevice42, StartAddress=0mb, Size=8656mb),

(PhysicalDevice43, StartAddress=0mb, Size=8656mb),

(PhysicalDevice44, StartAddress=0mb, Size=8656mb),

(PhysicalDevice45, StartAddress=0mb, Size=8656mb),

(PhysicalDevice46, StartAddress=0mb, Size=8656mb),

(PhysicalDevice47, StartAddress=0mb, Size=8656mb),

(PhysicalDevice48, StartAddress=0mb, Size=8656mb),

(PhysicalDevice49, StartAddress=0mb, Size=8656mb),

(PhysicalDevice50, StartAddress=0mb, Size=8656mb),

(PhysicalDevice51, StartAddress=0mb, Size=8656mb),

(PhysicalDevice52, StartAddress=0mb, Size=8656mb),

(PhysicalDevice53, StartAddress=0mb, Size=8656mb),

(PhysicalDevice54, StartAddress=0mb, Size=8656mb),

(PhysicalDevice55, StartAddress=0mb, Size=8656mb);

LogicalDevice0 = StripeSize=64kb, Raid=12, WriteThrough=1, Size=484736mb, BIOSGeometry=8GB,

(IntermediateDevice0, StartAddress=0mb, Size=121184mb),

(IntermediateDevice1, StartAddress=0mb, Size=121184mb),

(IntermediateDevice2, StartAddress=0mb, Size=121184mb),

(IntermediateDevice3, StartAddress=0mb, Size=121184mb);

EndGroup

BeginControllerParameter

ControllerName = eXtremeRAID 2000;

ControllerType = 28;

FirmwareVersion = 5.60;

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 = 1;

DisableBios = 0;

EnableCDROMBoot = 0;

EnableStorageWorks = 0;

EnableSAFTE = 0;

EnableSES = 0;

EnableARM = 0;

EnableOFM = 1;

OEMCode = 0;

StartupOption = 0;

EndControllerParameter

End

Mylex eXtremeRAID Adapter 2

Begin

BeginGroup

PhysicalDevice0 = Channel=0, Target=0, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice1 = Channel=0, Target=1, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice2 = Channel=0, Target=2, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice3 = Channel=0, Target=3, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice4 = Channel=0, Target=4, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice5 = Channel=0, Target=5, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice6 = Channel=0, Target=6, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice7 = Channel=0, Target=8, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice8 = Channel=0, Target=9, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice9 = Channel=0, Target=10, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice10 = Channel=0, Target=11, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice11 = Channel=0, Target=12, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice12 = Channel=0, Target=13, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice13 = Channel=0, Target=14, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice14 = Channel=1, Target=0, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice15 = Channel=1, Target=1, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice16 = Channel=1, Target=2, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice17 = Channel=1, Target=3, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice18 = Channel=1, Target=4, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice19 = Channel=1, Target=5, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice20 = Channel=1, Target=6, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice21 = Channel=1, Target=8, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice22 = Channel=1, Target=9, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice23 = Channel=1, Target=10, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice24 = Channel=1, Target=11, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice25 = Channel=1, Target=12, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice26 = Channel=1, Target=13, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice27 = Channel=1, Target=14, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice28 = Channel=2, Target=0, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice29 = Channel=2, Target=1, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice30 = Channel=2, Target=2, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice31 = Channel=2, Target=3, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice32 = Channel=2, Target=4, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice33 = Channel=2, Target=5, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice34 = Channel=2, Target=6, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice35 = Channel=2, Target=8, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice36 = Channel=2, Target=9, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice37 = Channel=2, Target=10, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice38 = Channel=2, Target=11, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice39 = Channel=2, Target=12, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice40 = Channel=2, Target=13, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice41 = Channel=2, Target=14, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice42 = Channel=3, Target=0, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice43 = Channel=3, Target=1, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice44 = Channel=3, Target=2, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice45 = Channel=3, Target=3, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice46 = Channel=3, Target=4, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice47 = Channel=3, Target=5, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice48 = Channel=3, Target=6, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice49 = Channel=3, Target=8, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

```
        PhysicalDevice50 = Channel=3, Target=9, Size=8662mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        PhysicalDevice51 = Channel=3, Target=10, Size=8662mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        PhysicalDevice52 = Channel=3, Target=11, Size=8662mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        PhysicalDevice53 = Channel=3, Target=12, Size=8662mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        PhysicalDevice54 = Channel=3, Target=13, Size=8662mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        PhysicalDevice55 = Channel=3, Target=14, Size=8662mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        IntermediateDevice0 = StripeSize=64kb, Raid=0, WriteThrough=1,
Size=121184mb,
                (PhysicalDevice0, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice1, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice2, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice3, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice4, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice5, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice6, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice7, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice8, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice9, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice10, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice11, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice12, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice13, StartAddress=0mb, Size=8656mb);
        IntermediateDevice1 = StripeSize=64kb, Raid=0, WriteThrough=1,
Size=121184mb,
                (PhysicalDevice14, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice15, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice16, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice17, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice18, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice19, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice20, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice21, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice22, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice23, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice24, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice25, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice26, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice27, StartAddress=0mb, Size=8656mb);
        IntermediateDevice2 = StripeSize=64kb, Raid=0, WriteThrough=1,
Size=121184mb,
                (PhysicalDevice28, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice29, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice30, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice31, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice32, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice33, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice34, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice35, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice36, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice37, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice38, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice39, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice40, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice41, StartAddress=0mb, Size=8656mb);
        IntermediateDevice3 = StripeSize=64kb, Raid=0, WriteThrough=1,
Size=121184mb,
                (PhysicalDevice42, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice43, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice44, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice45, StartAddress=0mb, Size=8656mb),
                (PhysicalDevice46, StartAddress=0mb, Size=8656mb),
```

(PhysicalDevice47, StartAddress=0mb, Size=8656mb),

(PhysicalDevice48, StartAddress=0mb, Size=8656mb),

(PhysicalDevice49, StartAddress=0mb, Size=8656mb),

(PhysicalDevice50, StartAddress=0mb, Size=8656mb),

(PhysicalDevice51, StartAddress=0mb, Size=8656mb),

(PhysicalDevice52, StartAddress=0mb, Size=8656mb),

(PhysicalDevice53, StartAddress=0mb, Size=8656mb),

(PhysicalDevice54, StartAddress=0mb, Size=8656mb),

(PhysicalDevice55, StartAddress=0mb, Size=8656mb);

LogicalDevice0 = StripeSize=64kb, Raid=12, WriteThrough=1, Size=484736mb, BIOSGeometry=8GB,

(IntermediateDevice0, StartAddress=0mb, Size=121184mb),

(IntermediateDevice1, StartAddress=0mb, Size=121184mb),

(IntermediateDevice2, StartAddress=0mb, Size=121184mb),

(IntermediateDevice3, StartAddress=0mb, Size=121184mb);

EndGroup

BeginControllerParameter

ControllerName = eXtremeRAID 2000;

ControllerType = 28;

FirmwareVersion = 5.60;

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 = 1;

DisableBios = 0;

EnableCDROMBoot = 0;

EnableStorageWorks = 0;

EnableSAFTE = 0;

EnableSES = 0;

EnableARM = 0;

EnableOFM = 1;

OEMCode = 0;

StartupOption = 0;

EndControllerParameter

End

Mylex eXtremeRAID Adapter 3

Begin

BeginGroup

PhysicalDevice0 = Channel=0, Target=0, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice1 = Channel=0, Target=1, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice2 = Channel=0, Target=2, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice3 = Channel=0, Target=3, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice4 = Channel=0, Target=4, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice5 = Channel=0, Target=5, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice6 = Channel=0, Target=6, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice7 = Channel=0, Target=8, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice8 = Channel=0, Target=9, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice9 = Channel=0, Target=10, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice10 = Channel=0, Target=11, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice11 = Channel=0, Target=12, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice12 = Channel=0, Target=13, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice13 = Channel=0, Target=14, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice14 = Channel=1, Target=0, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice15 = Channel=1, Target=1, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice16 = Channel=1, Target=2, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice17 = Channel=1, Target=3, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice18 = Channel=1, Target=4, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice19 = Channel=1, Target=5, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice20 = Channel=1, Target=6, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice21 = Channel=1, Target=8, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice22 = Channel=1, Target=9, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice23 = Channel=1, Target=10, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice24 = Channel=1, Target=11, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice25 = Channel=1, Target=12, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice26 = Channel=1, Target=13, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice27 = Channel=1, Target=14, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice28 = Channel=2, Target=0, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice29 = Channel=2, Target=1, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice30 = Channel=2, Target=2, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice31 = Channel=2, Target=3, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice32 = Channel=2, Target=4, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice33 = Channel=2, Target=5, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice34 = Channel=2, Target=6, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice35 = Channel=2, Target=8, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice36 = Channel=2, Target=9, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice37 = Channel=2, Target=10, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice38 = Channel=2, Target=11, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice39 = Channel=2, Target=12, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice40 = Channel=2, Target=13, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice41 = Channel=2, Target=14, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice42 = Channel=3, Target=0, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice43 = Channel=3, Target=1, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice44 = Channel=3, Target=2, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice45 = Channel=3, Target=3, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice46 = Channel=3, Target=4, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice47 = Channel=3, Target=5, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice48 = Channel=3, Target=6, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice49 = Channel=3, Target=8, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice50 = Channel=3, Target=9, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice51 = Channel=3, Target=10, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice52 = Channel=3, Target=11, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice53 = Channel=3, Target=12, Size=8662mb,
State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

PhysicalDevice54 = Channel=3, Target=13, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

PhysicalDevice55 = Channel=3, Target=14, Size=8662mb, State=Online,

TransferSpeed=80MHz, TransferWidth=16Bit, MaxTag=16;

IntermediateDevice0 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=121184mb,

(PhysicalDevice0, StartAddress=0mb, Size=8656mb),

(PhysicalDevice1, StartAddress=0mb, Size=8656mb),

(PhysicalDevice2, StartAddress=0mb, Size=8656mb),

(PhysicalDevice3, StartAddress=0mb, Size=8656mb),

(PhysicalDevice4, StartAddress=0mb, Size=8656mb),

(PhysicalDevice5, StartAddress=0mb, Size=8656mb),

(PhysicalDevice6, StartAddress=0mb, Size=8656mb),

(PhysicalDevice7, StartAddress=0mb, Size=8656mb),

(PhysicalDevice8, StartAddress=0mb, Size=8656mb),

(PhysicalDevice9, StartAddress=0mb, Size=8656mb),

(PhysicalDevice10, StartAddress=0mb, Size=8656mb),

(PhysicalDevice11, StartAddress=0mb, Size=8656mb),

(PhysicalDevice12, StartAddress=0mb, Size=8656mb),

(PhysicalDevice13, StartAddress=0mb, Size=8656mb);

IntermediateDevice1 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=121184mb,

(PhysicalDevice14, StartAddress=0mb, Size=8656mb),

(PhysicalDevice15, StartAddress=0mb, Size=8656mb),

(PhysicalDevice16, StartAddress=0mb, Size=8656mb),

(PhysicalDevice17, StartAddress=0mb, Size=8656mb),

(PhysicalDevice18, StartAddress=0mb, Size=8656mb),

(PhysicalDevice19, StartAddress=0mb, Size=8656mb),

(PhysicalDevice20, StartAddress=0mb, Size=8656mb),

(PhysicalDevice21, StartAddress=0mb, Size=8656mb),

(PhysicalDevice22, StartAddress=0mb, Size=8656mb),

(PhysicalDevice23, StartAddress=0mb, Size=8656mb),

(PhysicalDevice24, StartAddress=0mb, Size=8656mb),

(PhysicalDevice25, StartAddress=0mb, Size=8656mb),

(PhysicalDevice26, StartAddress=0mb, Size=8656mb),

(PhysicalDevice27, StartAddress=0mb, Size=8656mb);

IntermediateDevice2 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=121184mb,

(PhysicalDevice28, StartAddress=0mb, Size=8656mb),

(PhysicalDevice29, StartAddress=0mb, Size=8656mb),

(PhysicalDevice30, StartAddress=0mb, Size=8656mb),

(PhysicalDevice31, StartAddress=0mb, Size=8656mb),

(PhysicalDevice32, StartAddress=0mb, Size=8656mb),

(PhysicalDevice33, StartAddress=0mb, Size=8656mb),

(PhysicalDevice34, StartAddress=0mb, Size=8656mb),

(PhysicalDevice35, StartAddress=0mb, Size=8656mb),

(PhysicalDevice36, StartAddress=0mb, Size=8656mb),

(PhysicalDevice37, StartAddress=0mb, Size=8656mb),

(PhysicalDevice38, StartAddress=0mb, Size=8656mb),

(PhysicalDevice39, StartAddress=0mb, Size=8656mb),

(PhysicalDevice40, StartAddress=0mb, Size=8656mb),

(PhysicalDevice41, StartAddress=0mb, Size=8656mb);

IntermediateDevice3 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=121184mb,

(PhysicalDevice42, StartAddress=0mb, Size=8656mb),

(PhysicalDevice43, StartAddress=0mb, Size=8656mb),

(PhysicalDevice44, StartAddress=0mb, Size=8656mb),

(PhysicalDevice45, StartAddress=0mb, Size=8656mb),

(PhysicalDevice46, StartAddress=0mb, Size=8656mb),

(PhysicalDevice47, StartAddress=0mb, Size=8656mb),

(PhysicalDevice48, StartAddress=0mb, Size=8656mb),

(PhysicalDevice49, StartAddress=0mb, Size=8656mb),

(PhysicalDevice50, StartAddress=0mb, Size=8656mb),

(PhysicalDevice51, StartAddress=0mb, Size=8656mb),

(PhysicalDevice52, StartAddress=0mb, Size=8656mb),

(PhysicalDevice53, StartAddress=0mb, Size=8656mb),

(PhysicalDevice54, StartAddress=0mb, Size=8656mb),

(PhysicalDevice55, StartAddress=0mb, Size=8656mb);

LogicalDevice0 = StripeSize=64kb, Raid=12, WriteThrough=1, Size=484736mb, BIOSGeometry=8GB,

(IntermediateDevice0, StartAddress=0mb, Size=121184mb),

```
                (IntermediateDevice1, StartAddress=0mb,
Size=121184mb),
                (IntermediateDevice2, StartAddress=0mb,
Size=121184mb),
                (IntermediateDevice3, StartAddress=0mb,
Size=121184mb);
 EndGroup
 BeginControllerParameter
        ControllerName = eXtremeRAID 2000;
        ControllerType = 28;
        FirmwareVersion = 5.60;
        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 = 0;
        EnableCDROMBoot = 0;
        EnableStorageWorks = 0;
        EnableSAFTE = 0;
        EnableSES = 0;
        EnableARM = 1;
        EnableOFM = 1;
        OEMCode = 0;
        StartupOption = 0;
 EndControllerParameter
End
```

Mylex eXtremeRAID Adapter 4

```
Begin
 BeginGroup
        PhysicalDevice0 = Channel=0, Target=0, Size=17340mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        PhysicalDevice1 = Channel=1, Target=8, Size=17340mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        PhysicalDevice2 = Channel=0, Target=1, Size=17340mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        PhysicalDevice3 = Channel=1, Target=9, Size=17340mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        PhysicalDevice4 = Channel=0, Target=2, Size=17340mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        PhysicalDevice5 = Channel=1, Target=10, Size=17340mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        PhysicalDevice6 = Channel=0, Target=3, Size=17340mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        PhysicalDevice7 = Channel=1, Target=11, Size=17340mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        PhysicalDevice8 = Channel=0, Target=4, Size=17340mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        PhysicalDevice9 = Channel=1, Target=12, Size=17340mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
        PhysicalDevice10 = Channel=0, Target=5, Size=17340mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;
```

```
        PhysicalDevice11 = Channel=1, Target=13, Size=17340mb,
State=Online,
                TransferSpeed=80MHz, TransferWidth=16Bit,
MaxTag=16;

        IntermediateDevice0 = StripeSize=64kb, Raid=1, WriteThrough=1,
Size=17336mb,
                (PhysicalDevice0, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice1, StartAddress=0mb, Size=17336mb);

        IntermediateDevice1 = StripeSize=64kb, Raid=1, WriteThrough=1,
Size=17336mb,
                (PhysicalDevice2, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice3, StartAddress=0mb, Size=17336mb);

        IntermediateDevice2 = StripeSize=64kb, Raid=1, WriteThrough=1,
Size=17336mb,
                (PhysicalDevice4, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice5, StartAddress=0mb, Size=17336mb);

        IntermediateDevice3 = StripeSize=64kb, Raid=1, WriteThrough=1,
Size=17336mb,
                (PhysicalDevice6, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice7, StartAddress=0mb, Size=17336mb);

        IntermediateDevice4 = StripeSize=64kb, Raid=1, WriteThrough=1,
Size=17336mb,
                (PhysicalDevice8, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice9, StartAddress=0mb, Size=17336mb);

        IntermediateDevice5 = StripeSize=64kb, Raid=1, WriteThrough=1,
Size=17336mb,
                (PhysicalDevice10, StartAddress=0mb, Size=17336mb),
                (PhysicalDevice11, StartAddress=0mb, Size=17336mb);

        LogicalDevice0 = StripeSize=64kb, Raid=12, WriteThrough=1,
Size=104016mb, BIOSGeometry=8GB,
                (IntermediateDevice0, StartAddress=0mb,
Size=17336mb),
                (IntermediateDevice1, StartAddress=0mb,
Size=17336mb),
                (IntermediateDevice2, StartAddress=0mb,
Size=17336mb),
                (IntermediateDevice3, StartAddress=0mb,
Size=17336mb),
                (IntermediateDevice4, StartAddress=0mb,
Size=17336mb),
                (IntermediateDevice5, StartAddress=0mb,
Size=17336mb);

 EndGroup

BeginControllerParameter
        ControllerName = eXtremeRAID 2000;
        ControllerType = 28;
        FirmwareVersion = 5.60;
        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 = 1;
        DisableBios = 0;
        EnableCDROMBoot = 0;
        EnableStorageWorks = 0;
        EnableSAFTE = 0;
        EnableSES = 0;
        EnableARM = 0;
        EnableOFM = 1;
        OEMCode = 0;
        StartupOption = 0;
 EndControllerParameter

End
```

Client Configuration Parameters

Microsoft Windows 2000 Server Configuration Parameters

```
System Information report written at: 11/27/2000 11:25:33 AM
[System Information]

[ Following are sub-categories of this main category ]

[System Summary]
```

Item	Value
OS Name	Microsoft Windows 2000 Server
Version	5.0.2195 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	SQLCLIENT10
System Manufacturer	IBM
System Model	-[86582RY]-
System Type	X86-based PC
Processor	x86 Family 6 Model 8 Stepping 3 GenuineIntel ~731 Mhz
BIOS Version	IBM BIOS Ver 6.0
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition1
Locale	United States
User Name	SQLCLIENT10\Administrator
Time Zone	Eastern Standard Time
Total Physical Memory	392,728 KB
Available Physical Memory	297,252 KB
Total Virtual Memory	1,533,656 KB
Available Virtual Memory	1,373,664 KB
Page File Space	1,140,928 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
0x0000-0x0CF7	PCI bus	OK
0x0000-0x0CF7	Direct memory access controller	OK
0x0D00-0x4AEB	PCI bus	OK
0x82E8-0xFFFF	PCI bus	OK
0x03B0-0x03BB	S3 Inc. Savage4	OK
0x03C0-0x03DF	S3 Inc. Savage4	OK
0x2000-0x201F	IBM 10/100 NetFinity Fault Tolerant Adapter	OK
0x0A79-0x0A79	ISAPNP Read Data Port	OK
0x0279-0x0279	ISAPNP Read Data Port	OK
0x02F4-0x02F7	ISAPNP Read Data Port	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
0x04D0-0x04D1	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
0x0F50-0x0F58	Motherboard resources	OK
0xFD00-0xFD3F	Motherboard resources	OK
0xFE80-0xFEBF	Motherboard resources	OK
0xFEC0-0xFEDF	Motherboard resources	OK
0xEE9B-0xEE9B	Motherboard resources	OK
0x00E8-0x00E9	Not Available	OK
0x0840-0x084F	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
0x4AEC-0x82E7	PCI bus	OK
0x4B00-0x4BFF	Adaptec AIC-7899 Ultra160/m PCI SCSI Card	OK
0x4C00-0x4CFF	Adaptec AIC-7899 Ultra160/m PCI SCSI Card	OK
0x4D00-0x4D1F	IBM 10/100 EtherJet PCI Adapter	OK

[IRQs]

IRQ Number	Device
30	Microsoft ACPI-Compliant System
31	S3 Inc. Savage4
27	IBM 10/100 NetFinity Fault Tolerant Adapter
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
5	Not Available
14	Primary IDE Channel
11	Standard OpenHCD USB Host Controller
28	Adaptec AIC-7899 Ultra160/m PCI SCSI Card
29	Adaptec AIC-7899 Ultra160/m PCI SCSI Card
20	IBM 10/100 EtherJet PCI Adapter

[Memory]

Range	Device	Status
0xC8000-0xCFFFF	PCI bus	OK
0xC8000-0xCFFFF	System board	OK
0xF0000000-0xFEBFFFFF	PCI bus	OK
0xF0000000-0xFEBFFFFF	S3 Inc. Savage4	OK
0xFED00000-0xFEDFFFFF	PCI bus	OK
0xFEF00000-0xFFDFFFFF	PCI bus	OK
0xFEB80000-0xFEBFFFFF	S3 Inc. Savage4	OK
0xA0000-0xBFFFF	S3 Inc. Savage4	OK
0xFEB7FC00-0xFEB7FC1F	IBM 10/100 NetFinity Fault Tolerant Adapter	OK
0xFEC00000-0xFECFFFFF	Advanced programmable interrupt controller	OK
0xFEE00000-0xFEEFFFFF	Advanced programmable interrupt controller	OK
0xFF700000-0xFF700FFF	Standard OpenHCD USB Host Controller	OK
0xD0000-0xDFFFF	PCI bus	OK
0x18000000-0xEFFFFFFF	PCI bus	OK
0xEFFFF000-0xEFFFFFFF	Adaptec AIC-7899 Ultra160/m PCI SCSI Card	OK
0xEFFFE000-0xEFFFEFFF	Adaptec AIC-7899 Ultra160/m PCI SCSI Card	OK
0xEFFFD000-0xEFFFDFFF	IBM 10/100 EtherJet PCI Adapter	OK
0xEFE00000-0xEFEFFFFF	IBM 10/100 EtherJet PCI Adapter	OK
0x0000-0x9FFFF	System board	OK

0x100000-0xFFFFFF System board OK
0x1000000-0x17FFFFF System board OK
0xE0000-0xFFFFF System board OK
0xFFFE0000-0xFFFFFFFF System board OK
0xCC000-0xCFFFF System board 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 7:00:00 AM
c:\winnt\system32\msg723.acm Microsoft Corporation OK C:\WINNT\System32\MSG723.ACM 4.4.3385 106.77 KB (109,328 bytes) 10/4/2000 4:32:18 PM
c:\winnt\system32\lhacm.acm Microsoft Corporation OK C:\WINNT\System32\LHACM.ACM 4.4.3385 33.27 KB (34,064 bytes) 10/4/2000 4:32:19 PM
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 7:00:00 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 7:00:00 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 7:00:00 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 7:00:00 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 7:00:00 AM

[Video Codecs]

Codec Manufacturer Description Status File Version Size Creation Date
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\msh261.drv Microsoft Corporation OK C:\WINNT\System32\MSH261.DRV 4.4.3385 163.77 KB (167,696 bytes) 10/4/2000 4:32:19 PM
c:\winnt\system32\msh263.drv Microsoft Corporation OK C:\WINNT\System32\MSH263.DRV 4.4.3385 252.27 KB (258,320 bytes) 10/4/2000 4:31:49 PM
c:\winnt\system32\msvidc32.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\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\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
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

[CD-ROM]

Item Value
Drive D:
Description CD-ROM Drive
Media Loaded False
Media Type CD-ROM
Name LG CD-ROM CRD-8481B
Manufacturer (Standard CD-ROM drives)
Status OK
Transfer Rate Not Available
SCSI Target ID 0
PNP Device ID
IDE\CDROMLG_CD-ROM_CRD-8481B_____1.01____\30323030302F2F323932202020202020202020

[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_04\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 800 x 600 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&F0B8F99&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&F0B8F99&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] IBM 10/100 NetFinity Fault Tolerant Adapter
Adapter Type Ethernet 802.3
Product Name IBM 10/100 NetFinity Fault Tolerant Adapter
Installed True
PNP Device ID
PCI\VEN_1022&DEV_2000&SUBSYS_20001014&REV_44\3&267A616A&0&10
Last Reset 11/27/2000 6:21:10 AM
Index 0
Service Name PCnet
IP Address 192.168.125.10
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 00:06:29:D5:7C:E9
Service Name PCnet
IRQ Number 27
I/O Port 0x2000-0x201F
Driver c:\winnt\system32\drivers\pcntn5m.sys (33811, 4.23.00)

Name [00000001] Intel(R) PRO/100+ Dual Port Server Adapter
Adapter Type Not Available
Product Name Intel(R) PRO/100+ Dual Port Server Adapter
Installed True
PNP Device ID Not Available
Last Reset 11/27/2000 6:21:10 AM
Index 1
Service Name E100B
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled True
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Not Available

Name [00000002] Intel(R) PRO/100+ Dual Port Server Adapter
Adapter Type Not Available
Product Name Intel(R) PRO/100+ Dual Port Server Adapter
Installed True
PNP Device ID Not Available
Last Reset 11/27/2000 6:21:10 AM
Index 2
Service Name E100B
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled True
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Not Available

Name [00000003] Intel(R) PRO/100+ Dual Port Server Adapter
Adapter Type Not Available
Product Name Intel(R) PRO/100+ Dual Port Server Adapter
Installed True
PNP Device ID Not Available
Last Reset 11/27/2000 6:21:10 AM
Index 3
Service Name E100B
IP Address 192.168.125.10
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled True
DHCP Server
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:06:29:D5:7C:E9
Service Name Not Available

Name [00000004] Intel(R) PRO/100+ Dual Port Server Adapter
Adapter Type Not Available
Product Name Intel(R) PRO/100+ Dual Port Server Adapter
Installed True
PNP Device ID Not Available
Last Reset 11/27/2000 6:21:10 AM
Index 4
Service Name E100B
IP Address 192.168.10.99
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled True
DHCP Server
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:04:AC:36:44:EC
Service Name Not Available

Name [00000005] Intel(R) PRO/100+ Dual Port Server Adapter
Adapter Type Not Available
Product Name Intel(R) PRO/100+ Dual Port Server Adapter
Installed True
PNP Device ID Not Available
Last Reset 11/27/2000 6:21:10 AM
Index 5
Service Name E100B
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled True
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Not Available

Name [00000006] Intel(R) PRO/100+ Dual Port Server Adapter
Adapter Type Not Available
Product Name Intel(R) PRO/100+ Dual Port Server Adapter
Installed True
PNP Device ID Not Available
Last Reset 11/27/2000 6:21:10 AM
Index 6
Service Name E100B
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 [00000007] RAS Async Adapter
Adapter Type Not Available
Product Name RAS Async Adapter
Installed True
PNP Device ID Not Available
Last Reset 11/27/2000 6:21:10 AM
Index 7
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 [00000008] WAN Miniport (L2TP)
Adapter Type Not Available
Product Name WAN Miniport (L2TP)
Installed True
PNP Device ID ROOT\MS_L2TPMINIPORT\0000
Last Reset 11/27/2000 6:21:10 AM
Index 8
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 [00000009] 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 11/27/2000 6:21:10 AM
Index 9
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\raspptp.sys (47856, 5.00.2160.1)

Name [00000010] Direct Parallel
Adapter Type Not Available
Product Name Direct Parallel
Installed True
PNP Device ID ROOT\MS_PTIMINIPORT\0000
Last Reset 11/27/2000 6:21:10 AM
Index 10
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 [00000011] WAN Miniport (IP)
Adapter Type Not Available
Product Name WAN Miniport (IP)
Installed True
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 11/27/2000 6:21:10 AM
Index 11
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 (90768, 5.00.2184.1)

Name [00000012] IBM 10/100 EtherJet PCI Adapter
Adapter Type Ethernet 802.3
Product Name IBM 10/100 EtherJet PCI Adapter
Installed True
PNP Device ID
PCI\VEN_8086&DEV_1229&SUBSYS_005C1014&REV_05\3&13C0B0C5&0&28
Last Reset 11/27/2000 6:21:10 AM
Index 12
Service Name IBMFE
IP Address 192.168.10.99
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 00:04:AC:36:44:EC
Service Name IBMFE
IRQ Number 20
I/O Port 0x4D00-0x4D1F
Driver c:\winnt\system32\drivers\ibmfent5.sys (80656, 4.01.75.0000)

Name [00000013] IBM 10/100 EtherJet PCI Adapter
Adapter Type Not Available
Product Name IBM 10/100 EtherJet PCI Adapter
Installed True
PNP Device ID Not Available
Last Reset 11/27/2000 6:21:10 AM
Index 13
Service Name IBMFE
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 [00000014] IBM 10/100 EtherJet PCI Adapter
Adapter Type Not Available
Product Name IBM 10/100 EtherJet PCI Adapter
Installed True
PNP Device ID Not Available
Last Reset 11/27/2000 6:21:10 AM

Index 14
Service Name IBMFE
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

[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_{519DF164-C2E2-4B95-ADCA-A5FE91C070D0}] SEQPACKET 11
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_{519DF164-C2E2-4B95-ADCA-A5FE91C070D0}] DATAGRAM 11
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_{C04567B4-9B0B-4A29-8279-471F5F20794E}] SEQPACKET 10
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_{C04567B4-9B0B-4A29-8279-471F5F20794E}] DATAGRAM 10
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_{10FE3B77-2ED7-4549-B020-0F3082ED2016}] SEQPACKET 9
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_{10FE3B77-2ED7-4549-B020-0F3082ED2016}] DATAGRAM 9
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_{D074967D-9B75-4E93-8CEA-7349F1DB5580}] SEQPACKET 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_{D074967D-9B75-4E93-8CEA-7349F1DB5580}] 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_{B42D6592-9A44-4456-9C18-C04769F97685}] 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_{B42D6592-9A44-4456-9C18-C04769F97685}] 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_{77DB4B13-E43C-40D8-B180-7447B4352EF1}] 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_{77DB4B13-E43C-40D8-B180-7447B4352EF1}] 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_{7359D11B-7E1A-411C-A883-53C42F37D3D3}] 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_{7359D11B-7E1A-411C-A883-53C42F37D3D3}] 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

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{5463FFD7-5102-466B-B730-BE6C592F2580}] SEQPACKET 4
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_{5463FFD7-5102-466B-B730-BE6C592F2580}] DATAGRAM 4
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_{F9202C21-9BBB-489B-B286-1D59C7ACC80C}] SEQPACKET 5
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_{F9202C21-9BBB-489B-B286-1D59C7ACC80C}] DATAGRAM 5
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_{B1137635-649F-4AF6-8588-EAD18CD35523}] 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_{B1137635-649F-4AF6-8588-EAD18CD35523}] DATAGRAM 6
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_{F422FA62-9410-46F2-B308-66433B4E0553}] SEQPACKET 7
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_{F422FA62-9410-46F2-B308-66433B4E0553}] DATAGRAM 7
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_{99DC9EBE-437C-4834-8841-32B09B811C94}] SEQPACKET 8
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_{99DC9EBE-437C-4834-8841-32B09B811C94}] DATAGRAM 8
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	5.00.2152.1
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 (62448, 5.00.2134.1)

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 (62448, 5.00.2134.1)

[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	6.89 GB (7,395,287,040 bytes)
Volume Name	
Volume Serial Number	E4DE2F60
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 SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 2
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 9097159680 bytes
Drive TotalCylinders 1106
Drive TotalSectors 17767890
Drive TotalTracks 282030
Drive TracksPerCylinder 255

[SCSI]

Item Value
Name Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Caption Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Driver adpu160m
Status OK
PNP Device ID PCI\VEN_9005&DEV_00CF&SUBSYS_019D1014&REV_01\3&13C0B0C5&0&18
Device ID PCI\VEN_9005&DEV_00CF&SUBSYS_019D1014&REV_01\3&13C0B0C5&0&18
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 28
I/O Port 0x4B00-0x4BFF
Driver c:\winnt\system32\drivers\adpu160m.sys (64432, v3.10a)

Name Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Caption Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Driver adpu160m
Status OK
PNP Device ID PCI\VEN_9005&DEV_00CF&SUBSYS_019D1014&REV_01\3&13C0B0C5&0&19
Device ID PCI\VEN_9005&DEV_00CF&SUBSYS_019D1014&REV_01\3&13C0B0C5&0&19
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 29
I/O Port 0x4C00-0x4CFF
Driver c:\winnt\system32\drivers\adpu160m.sys (64432, v3.10a)

[Printing]

Name Port Name Server Name
No printing information

[Problem Devices]

Device PNP Device ID Error Code
Not Available ACPI\IBM37C0\4&F0B8F99&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 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
asyncmac RAS Asynchronous Media Driver c:\winnt\system32\drivers\asyncmac.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 False
deckzpsx deckzpsx Not Available Kernel Driver False
Disabled Stopped OK Normal False False
dfsdriver DfsDriver c:\winnt\system32\drivers\dfs.sys File System Driver
True Boot Running OK Normal 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 Kernel
Driver True Boot Running OK Normal False
True
dmboot dmboot c:\winnt\system32\drivers\dmboot.sys Kernel
Driver False Disabled Stopped OK Normal False
False
dmio Logical Disk Manager Driver c:\winnt\system32\drivers\dmio.sys
Kernel Driver True Boot Running OK Normal
False True
dmload dmload c:\winnt\system32\drivers\dmload.sys Kernel
Driver True Boot Running OK Normal False
True
e100b Intel PRO Adapter Driver
c:\winnt\system32\drivers\e100bnt5.sys Kernel Driver False
Manual Stopped OK Normal False False
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 File System
Driver True Disabled Running OK Normal False
True
fd16_700 Fd16_700 Not Available Kernel Driver False
Disabled Stopped OK Normal False False
fdc Floppy Disk Controller Driver c:\winnt\system32\drivers\fdc.sys
Kernel Driver True Manual Running OK Normal
False True
fireport fireport Not Available Kernel Driver False
Disabled Stopped OK Normal False False
flashpnt flashpnt Not Available Kernel Driver False
Disabled Stopped OK Normal False False
flpydisk Floppy Disk Driver c:\winnt\system32\drivers\flpydisk.sys
Kernel Driver True Manual Running OK Normal
False True
ftdisk Volume Manager Driver c:\winnt\system32\drivers\ftdisk.sys
Kernel Driver True Boot Running OK Normal
False True
gpc Generic Packet Classifier c:\winnt\system32\drivers\msgpc.sys
Kernel Driver True Manual Running OK Normal
False True
i8042prt i8042 Keyboard and PS/2 Mouse Port Driver
c:\winnt\system32\drivers\i8042prt.sys Kernel Driver True
System Running OK Normal False True
ibmfe IBM 10/100 Ethernet PCI Adapter Driver
c:\winnt\system32\drivers\ibmfent5.sys Kernel Driver True
Manual Running OK Normal False True
ini910u ini910u Not Available Kernel Driver False
Disabled Stopped OK Normal False False
intelide IntelIde Not Available Kernel Driver False
Disabled Stopped OK Normal False False
ipfilterdriver IP Traffic Filter Driver
c:\winnt\system32\drivers\ipfltdrv.sys Kernel Driver False
Manual Stopped OK Normal False False
ipinip IP in IP Tunnel Driver c:\winnt\system32\drivers\ipinip.sys
Kernel Driver False Manual Stopped OK Normal
False False
ipnat IP Network Address Translator c:\winnt\system32\drivers\ipnat.sys
Kernel Driver False Manual Stopped OK Normal
False False
ipsec IPSEC driver c:\winnt\system32\drivers\ipsec.sys
Kernel Driver True Manual Running OK Normal
False True
ipsraidn ipsraidn Not Available Kernel Driver False
Disabled Stopped OK Normal False False
isapnp PnP ISA/EISA Bus Driver c:\winnt\system32\drivers\isapnp.sys
Kernel Driver True Boot Running OK Critical
False True
kbdclass Keyboard Class Driver c:\winnt\system32\drivers\kbdclass.sys
Kernel Driver True System Running OK Normal
False True
ksecdd KSecDD c:\winnt\system32\drivers\ksecdd.sys Kernel
Driver True Boot Running OK Normal False
True
lbrtfdc lbrtfdc Not Available Kernel Driver False
System Stopped OK Ignore False False
lp6nds35 lp6nds35 Not Available Kernel Driver False
Disabled Stopped OK Normal False False
mnmdd mnmdd c:\winnt\system32\drivers\mnmdd.sys Kernel
Driver True System Running OK Ignore False
True
modem Modem c:\winnt\system32\drivers\modem.sys Kernel
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
False True
mountmgr MountMgr c:\winnt\system32\drivers\mountmgr.sys Kernel
Driver True Boot Running OK Normal False
True
mraid35x mraid35x Not Available Kernel Driver False
Disabled Stopped OK Normal False False
mrxsmb MRXSMB c:\winnt\system32\drivers\mrxsmb.sys File System
Driver True System Running OK Normal False
True
msfs Msfs c:\winnt\system32\drivers\msfs.sys File System Driver
True System Running OK Normal False True
mskssrv Microsoft Streaming Service Proxy
c:\winnt\system32\drivers\mskssrv.sys Kernel Driver False
Manual Stopped OK Normal False False
mspclock Microsoft Streaming Clock Proxy
c:\winnt\system32\drivers\mspclock.sys Kernel Driver False
Manual Stopped OK Normal False False
mspqm Microsoft Streaming Quality Manager Proxy
c:\winnt\system32\drivers\mspqm.sys Kernel Driver False
Manual Stopped OK Normal False False
mup Mup c:\winnt\system32\drivers\mup.sys File System Driver
True Boot Running OK Normal False True
ncrc710 Ncrc710 Not Available Kernel Driver False
Disabled Stopped OK Normal False False
ndis NDIS System Driver c:\winnt\system32\drivers\ndis.sys Kernel
Driver True Boot Running OK Normal False
True
ndistapi Remote Access NDIS TAPI Driver
c:\winnt\system32\drivers\ndistapi.sys Kernel Driver True
Manual Running OK Normal False True
ndiswan Remote Access NDIS WAN Driver
c:\winnt\system32\drivers\ndiswan.sys Kernel Driver True
Manual Running OK Normal False True
ndproxy NDIS Proxy c:\winnt\system32\drivers\ndproxy.sys
Kernel Driver True Manual Running OK Normal
False True
netbios NetBIOS Interface c:\winnt\system32\drivers\netbios.sys
File System Driver True System Running OK Normal
False True
netbt NetBios over Tcpip c:\winnt\system32\drivers\netbt.sys
Kernel Driver True System Running OK Normal
False True

netdetect NetDetect c:\winnt\system32\drivers\netdtect.sys Kernel
Driver False Manual Stopped OK Normal False
False
npfs Npfs c:\winnt\system32\drivers\npfs.sys File System Driver
True System Running OK Normal False True
ntfs Ntfs c:\winnt\system32\drivers\ntfs.sys File System Driver
True Disabled Running OK Normal False True
null Null c:\winnt\system32\drivers\null.sys Kernel Driver
True System Running OK Normal False True
nwlnkflt IPX Traffic Filter Driver
c:\winnt\system32\drivers\nwlnkflt.sys Kernel Driver False
Manual Stopped OK Normal False False
nwlnkfwd IPX Traffic Forwarder Driver
c:\winnt\system32\drivers\nwlnkfwd.sys Kernel Driver False
Manual Stopped OK Normal False False
openhci Microsoft USB Open Host Controller Driver
c:\winnt\system32\drivers\openhci.sys Kernel Driver True
Manual Running OK Normal False True
parallel Parallel class driver c:\winnt\system32\drivers\parallel.sys
Kernel Driver True Manual Running OK Normal
False True
parport Parallel port driver c:\winnt\system32\drivers\parport.sys
Kernel Driver True System Running OK Ignore
False True
partmgr PartMgr c:\winnt\system32\drivers\partmgr.sys Kernel
Driver True Boot Running OK Normal False
True
parvdm ParVdm c:\winnt\system32\drivers\parvdm.sys Kernel
Driver True Auto Running OK Ignore False
True
pci PCI Bus Driver c:\winnt\system32\drivers\pci.sys Kernel
Driver True Boot Running OK Critical False
True
pcidump PCIDump Not Available Kernel Driver False
System Stopped OK Ignore False False
pciide PCIIde c:\winnt\system32\drivers\pciide.sys Kernel
Driver True Boot Running OK Normal False
True
pcmcia Pcmcia c:\winnt\system32\drivers\pcmcia.sys Kernel
Driver False Disabled Stopped OK Normal False
False
pcnet PCNET Adapter Driver
c:\winnt\system32\drivers\pcntn5m.sys Kernel Driver True
Manual Running OK Normal False True
pdcomp PDCOMP Not Available Kernel Driver False
Manual Stopped OK Ignore False False
pdframe PDFRAME Not Available Kernel Driver
False Manual Stopped OK Ignore False False
pdreli PDRELI Not Available Kernel Driver False
Manual Stopped OK Ignore False False
pdrframe PDRFRAME Not Available Kernel Driver
False Manual Stopped OK Ignore False False
pptpminiport WAN Miniport (PPTP)
c:\winnt\system32\drivers\raspptp.sys Kernel Driver True
Manual Running OK Normal False True
ptilink Direct Parallel Link Driver c:\winnt\system32\drivers\ptilink.sys
Kernel Driver True Manual Running OK Normal
False True
ql1080 ql1080 Not Available Kernel Driver False
Disabled Stopped OK Normal False False
ql10wnt Ql10wnt Not Available Kernel Driver False
Disabled Stopped OK Normal False False
ql1240 ql1240 Not Available Kernel Driver False
Disabled Stopped OK Normal False False
ql2100 ql2100 Not Available Kernel Driver False
Disabled Stopped OK Normal False False
rasacd Remote Access Auto Connection Driver
c:\winnt\system32\drivers\rasacd.sys Kernel Driver True
System Running OK Normal False True
rasl2tp WAN Miniport (L2TP)
c:\winnt\system32\drivers\rasl2tp.sys Kernel Driver True
Manual Running OK Normal False True
raspti Direct Parallel c:\winnt\system32\drivers\raspti.sys
Kernel Driver True Manual Running OK Normal
False True
rca Microsoft Streaming Network Raw Channel Access
c:\winnt\system32\drivers\rca.sys Kernel Driver False Manual
Stopped OK Normal False False
rdbss Rdbss c:\winnt\system32\drivers\rdbss.sys File System
Driver True System Running OK Normal False
True
rdpwd RDPWD c:\winnt\system32\drivers\rdpwd.sys Kernel
Driver False Manual Stopped OK Ignore False
False
redbook Digital CD Audio Playback Filter Driver
c:\winnt\system32\drivers\redbook.sys Kernel Driver False
System Stopped OK Normal False False
s3inc S3Inc c:\winnt\system32\drivers\s3sav4m.sys Kernel
Driver True Manual Running OK Ignore False
True
serenum Serenum Filter Driver c:\winnt\system32\drivers\serenum.sys
Kernel Driver True Manual Running OK Normal
False True
serial Serial port driver c:\winnt\system32\drivers\serial.sys
Kernel Driver True System Running OK Ignore
False True
sfloppy Sfloppy c:\winnt\system32\drivers\sfloppy.sys Kernel
Driver False System Stopped OK Ignore False
False
sglfb sglfb Not Available Kernel Driver False
System Stopped OK Normal False False
simbad Simbad Not Available Kernel Driver False
Disabled Stopped OK Normal False False
sparrow Sparrow Not Available Kernel Driver False
Disabled Stopped OK Normal False False
spud Special Purpose Utility Driver c:\winnt\system32\drivers\spud.sys
Kernel Driver True Manual Running OK Normal
False True
srv Srv c:\winnt\system32\drivers\srv.sys File System Driver
True Manual Running OK Normal False True
swenum Software Bus Driver c:\winnt\system32\drivers\swenum.sys
Kernel Driver True Manual Running OK Normal
False True
symc810 symc810 Not Available Kernel Driver False
Disabled Stopped OK Normal False False
symc8xx symc8xx Not Available Kernel Driver False
Disabled Stopped OK Normal False False
sym_hi sym_hi Not Available Kernel Driver False
Disabled Stopped OK Normal False False
tcpip TCP/IP Protocol Driver c:\winnt\system32\drivers\tcpip.sys
Kernel Driver True System Running OK Normal
False True
tdasync TDASYNC c:\winnt\system32\drivers\tdasync.sys
Kernel Driver False Manual Stopped OK Ignore
False False
tdipx TDIPX c:\winnt\system32\drivers\tdipx.sys Kernel
Driver False Manual Stopped OK Ignore False
False
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
tdspx TDSPX c:\winnt\system32\drivers\tdspx.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 False
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 False
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 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 False
Manual Stopped OK Ignore False False

[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 8 Stepping 3,
GenuineIntel <SYSTEM>
PROCESSOR_REVISION 0803 <SYSTEM>
NUMBER_OF_PROCESSORS 1 <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
SQLCLIENT10\Administrator
TMP %USERPROFILE%\Local Settings\Temp
SQLCLIENT10\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
No network connections information

[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 160 11 204800
1413120 11/27/2000 11:21:31 AM 5.00.2170.1 44.27 KB
(45,328 bytes) 12/7/1999 7:00:00 AM
csrss.exe Not Available 184 13 Not Available
Not Available 11/27/2000 11:21:35 AM Unknown Unknown
Unknown
winlogon.exe c:\winnt\system32\winlogon.exe 204 13
204800 1413120 11/27/2000 11:21:36 AM 5.00.2182.1
173.27 KB (177,424 bytes) 12/7/1999 7:00:00 AM
services.exe c:\winnt\system32\services.exe 232 9
204800 1413120 11/27/2000 11:21:37 AM 5.00.2134.1
86.77 KB (88,848 bytes) 12/7/1999 7:00:00 AM
lsass.exe c:\winnt\system32\lsass.exe 244 13 204800
1413120 11/27/2000 11:21:37 AM 5.00.2184.1 32.77 KB
(33,552 bytes) 12/7/1999 7:00:00 AM
svchost.exe c:\winnt\system32\svchost.exe 416 8 204800
1413120 11/27/2000 11:21:41 AM 5.00.2134.1 7.77 KB
(7,952 bytes) 12/7/1999 7:00:00 AM
spoolsv.exe c:\winnt\system32\spoolsv.exe 444 8 204800
1413120 11/27/2000 11:21:41 AM 5.00.2161.1 43.77 KB
(44,816 bytes) 10/4/2000 12:10:53 PM
msdtc.exe c:\winnt\system32\msdtc.exe 472 8 204800
1413120 11/27/2000 11:21:41 AM 1999.9.3421.3 6.77 KB
(6,928 bytes) 10/4/2000 12:27:00 PM
svchost.exe c:\winnt\system32\svchost.exe 580 8 204800
1413120 11/27/2000 11:21:43 AM 5.00.2134.1 7.77 KB
(7,952 bytes) 12/7/1999 7:00:00 AM
llssrv.exe c:\winnt\system32\llssrv.exe 608 9 204800
1413120 11/27/2000 11:21:44 AM 5.00.2167.1 114.27 KB
(117,008 bytes) 12/7/1999 7:00:00 AM
regsvc.exe c:\winnt\system32\regsvc.exe 664 8 204800
1413120 11/27/2000 11:21:44 AM 5.00.2155.1 65.27 KB
(66,832 bytes) 12/7/1999 7:00:00 AM
rsvp.exe c:\winnt\system32\rsvp.exe 696 8 204800
1413120 11/27/2000 11:21:45 AM 5.00.2167.1 172.77 KB
(176,912 bytes) 12/7/1999 7:00:00 AM
mstask.exe c:\winnt\system32\mstask.exe 712 8 204800
1413120 11/27/2000 11:21:45 AM 4.71.2137.1 115.27 KB
(118,032 bytes) 10/4/2000 4:32:07 PM
tcpsvcs.exe c:\winnt\system32\tcpsvcs.exe 824 8 204800
1413120 11/27/2000 11:21:52 AM 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 848 8
204800 1413120 11/27/2000 11:21:52 AM 1.50.1085.0001
188.05 KB (192,567 bytes) 12/7/1999 7:00:00 AM
inetinfo.exe c:\winnt\system32\inetsrv\inetinfo.exe 888 8
204800 1413120 11/27/2000 11:21:53 AM 5.00.0984 14.27 KB
(14,608 bytes) 10/4/2000 12:27:41 PM
dfssvc.exe c:\winnt\system32\dfssvc.exe 900 8 204800
1413120 11/27/2000 11:21:53 AM 5.00.2191.1 85.27 KB
(87,312 bytes) 12/7/1999 7:00:00 AM
svchost.exe c:\winnt\system32\svchost.exe 1132 8 204800
1413120 11/27/2000 11:22:14 AM 5.00.2134.1 7.77 KB
(7,952 bytes) 12/7/1999 7:00:00 AM
explorer.exe c:\winnt\explorer.exe 740 8 204800
1413120 11/27/2000 11:22:39 AM 5.00.2920.0000 232.77 KB
(238,352 bytes) 12/7/1999 7:00:00 AM
ibmmon.exe c:\winnt\system32\ibmmon.exe 1168 8
204800 1413120 11/27/2000 11:22:41 AM 1.11 28.50 KB
(29,184 bytes) 10/6/2000 3:15:41 PM
mmc.exe c:\winnt\system32\mmc.exe 872 8 204800
1413120 11/27/2000 11:24:30 AM 5.00.2153.1 589.27 KB
(603,408 bytes) 12/7/1999 7:00:00 AM

[Loaded Modules]

Name Version Size File Date Manufacturer Path
wbemprox.dll 1.50.1085.0001 40.05 KB (41,016 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\wbemprox.dll
faxshell.dll 5.00.2134.1 8.27 KB (8,464 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\faxshell.dll
msacm32.dll 5.00.2134.1 65.27 KB (66,832 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\msacm32.dll
avifil32.dll 5.00.2134.1 76.27 KB (78,096 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\avifil32.dll
msvfw32.dll 5.00.2134.1 113.77 KB (116,496 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\msvfw32.dll
docprop2.dll 5.00.2178.1 297.77 KB (304,912 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\docprop2.dll
mlang.dll 5.00.2920.0000 510.77 KB (523,024 bytes) 12/7/1999 7:00:00 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 5.00.2191.1 194.27 KB (198,928 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\adsnt.dll
dbghelp.dll 5.00.2195.1 159.27 KB (163,088 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\dbghelp.dll
localsec.dll 5.00.2134.1 227.27 KB (232,720 bytes) 12/7/1999 7:00:00 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 5.00.2134.1 287.27 KB (294,160 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\filemgmt.dll
pdh.dll 5.00.2174.1 143.27 KB (146,704 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\pdh.dll
smlogcfg.dll 5.00.2163.1 273.27 KB (279,824 bytes) 12/7/1999 7:00:00 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) 10/4/2000 4:32:15 PM Microsoft Corporation c:\program files\common files\microsoft shared\msinfo\msinfo32.dll
riched20.dll 5.30.23.1200 421.27 KB (431,376 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\riched20.dll
riched32.dll 5.00.2134.1 3.77 KB (3,856 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\riched32.dll
els.dll 5.00.2175.1 151.27 KB (154,896 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\els.dll
ntmsmgr.dll 1,0,0,1 427.77 KB (438,032 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation and HighGround Systems, Inc. c:\winnt\system32\ntmsmgr.dll
mmfutil.dll 1.50.1085.0000 32.06 KB (32,829 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation 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) 12/7/1999 7:00:00 AM Executive Software International, Inc. c:\winnt\system32\dfrgres.dll
dfrgsnap.dll 5.00.2150.1 41.77 KB (42,768 bytes) 12/7/1999 7:00:00 AM Executive Software International, Inc. c:\winnt\system32\dfrgsnap.dll
dmdskres.dll 2191.1.296.2 119.00 KB (121,856 bytes) 12/7/1999 7:00:00 AM Microsoft Corp., VERITAS Software c:\winnt\system32\dmdskres.dll
dmutil.dll 2191.1.296.2 41.77 KB (42,768 bytes) 12/7/1999 7:00:00 AM VERITAS Software Corp. c:\winnt\system32\dmutil.dll
ntmsapi.dll 5.00.1948.1 50.27 KB (51,472 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\ntmsapi.dll
dmdskmgr.dll 2191.1.296.2 158.77 KB (162,576 bytes) 12/7/1999 7:00:00 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
mmcndmgr.dll 5.00.2178.1 815.27 KB (834,832 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\mmcndmgr.dll
mmc.exe 5.00.2153.1 589.27 KB (603,408 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\mmc.exe
ibmmon.exe 1.11 28.50 KB (29,184 bytes) 10/6/2000 3:15:41 PM IBM Corporation c:\winnt\system32\ibmmon.exe
wininet.dll 5.00.2920.0000 456.77 KB (467,728 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\wininet.dll
linkinfo.dll 5.00.2134.1 15.77 KB (16,144 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\linkinfo.dll
msi.dll 1.10.1029.0 1.71 MB (1,794,320 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\msi.dll
powrprof.dll 5.00.2920.0000 13.27 KB (13,584 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\powrprof.dll
batmeter.dll 5.00.2920.0000 20.27 KB (20,752 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\batmeter.dll
stobject.dll 5.00.2144.1 81.77 KB (83,728 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\stobject.dll
webcheck.dll 5.00.2920.0000 251.77 KB (257,808 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\webcheck.dll
ntshrui.dll 5.00.2134.1 46.77 KB (47,888 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\ntshrui.dll
mydocs.dll 5.00.2920.0000 55.77 KB (57,104 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\mydocs.dll
browseui.dll 5.00.2920.0000 793.27 KB (812,304 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\browseui.dll
shdocvw.dll 5.00.2920.0000 1.05 MB (1,104,144 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\shdocvw.dll
explorer.exe 5.00.2920.0000 232.77 KB (238,352 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\explorer.exe
tapisrv.dll 5.00.2186.1 168.77 KB (172,816 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\tapisrv.dll
dfssvc.exe 5.00.2191.1 85.27 KB (87,312 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\dfssvc.exe
iislog.dll 5.00.0984 76.27 KB (78,096 bytes) 10/4/2000 12:27:41 PM Microsoft Corporation c:\winnt\system32\inetsrv\iislog.dll
ntlsapi.dll 5.00.2134.1 6.77 KB (6,928 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\ntlsapi.dll
httpext.dll 0.9.3939.9 418.27 KB (428,304 bytes) 10/4/2000 12:27:41 PM Microsoft Corporation c:\winnt\system32\inetsrv\httpext.dll
rpcproxy.dll 5.00.2176.1 16.27 KB (16,656 bytes) 10/4/2000 12:26:54 PM Microsoft Corporation c:\winnt\system32\rpcproxy\rpcproxy.dll
fpexedll.dll 4.0.2.3406 20.06 KB (20,541 bytes) 10/4/2000 12:29:43 PM Microsoft Corporation c:\program files\common files\microsoft shared\web server extensions\40\bin\fpexedll.dll
md5filt.dll 5.00.0984 32.77 KB (33,552 bytes) 10/4/2000 12:27:47 PM Microsoft Corporation c:\winnt\system32\inetsrv\md5filt.dll
gzip.dll 5.00.0984 30.27 KB (30,992 bytes) 10/4/2000 12:27:46 PM Microsoft Corporation c:\winnt\system32\inetsrv\gzip.dll
compfilt.dll 5.00.0984 22.27 KB (22,800 bytes) 10/4/2000 12:27:46 PM Microsoft Corporation c:\winnt\system32\inetsrv\compfilt.dll
sspifilt.dll 5.00.0984 43.27 KB (44,304 bytes) 10/4/2000 12:27:48 PM Microsoft Corporation c:\winnt\system32\inetsrv\sspifilt.dll
iscomlog.dll 5.00.0984 24.77 KB (25,360 bytes) 10/4/2000 12:27:41 PM Microsoft Corporation c:\winnt\system32\inetsrv\iscomlog.dll

lonsint.dll 5.00.0984 11.77 KB (12,048 bytes) 10/4/2000 12:27:42 PM Microsoft Corporationc:\winnt\system32\inetsrv\lonsint.dll
inetsloc.dll5.00.0984 20.27 KB (20,752 bytes) 10/4/2000 12:27:43 PM Microsoft Corporationc:\winnt\system32\inetsloc.dll
iisfecnv.dll5.00.0984 7.27 KB (7,440 bytes)10/4/2000 12:27:41 PM Microsoft Corporationc:\winnt\system32\inetsrv\iisfecnv.dll
isatq.dll 5.00.0984 61.27 KB (62,736 bytes) 10/4/2000 12:27:43 PM Microsoft Corporationc:\winnt\system32\inetsrv\isatq.dll
infocomm.dll 5.00.0984 234.27 KB (239,888 bytes) 10/4/2000 12:27:41 PM Microsoft Corporation c:\winnt\system32\inetsrv\infocomm.dll
w3svc.dll 5.00.0984 347.27 KB (355,600 bytes) 10/4/2000 12:27:48 PM Microsoft Corporationc:\winnt\system32\inetsrv\w3svc.dll
security.dll5.00.2154.1 5.77 KB (5,904 bytes)12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\security.dll
svcext.dll 5.00.0984 39.77 KB (40,720 bytes) 10/4/2000 12:27:42 PM Microsoft Corporationc:\winnt\system32\inetsrv\svcext.dll
admexs.dll 5.00.0984 27.77 KB (28,432 bytes) 10/4/2000 12:27:41 PM Microsoft Corporationc:\winnt\system32\inetsrv\admexs.dll
wamreg.dll5.00.0984 46.27 KB (47,376 bytes) 10/4/2000 12:27:48 PM Microsoft Corporationc:\winnt\system32\inetsrv\wamreg.dll
metadata.dll 5.00.0984 70.77 KB (72,464 bytes) 10/4/2000 12:27:42 PM Microsoft Corporation c:\winnt\system32\inetsrv\metadata.dll
iismap.dll 5.00.0984 56.27 KB (57,616 bytes) 10/4/2000 12:27:43 PM Microsoft Corporationc:\winnt\system32\iismap.dll
nsepm.dll 5.00.0984 43.27 KB (44,304 bytes) 10/4/2000 12:27:42 PM Microsoft Corporationc:\winnt\system32\inetsrv\nsepm.dll
coadmin.dll 5.00.0984 39.77 KB (40,720 bytes) 10/4/2000 12:27:43 PM Microsoft Corporation c:\winnt\system32\inetsrv\coadmin.dll
iisadmin.dll 5.00.0984 14.77 KB (15,120 bytes) 10/4/2000 12:27:41 PM Microsoft Corporation c:\winnt\system32\inetsrv\iisadmin.dll
rpcref.dll 5.00.0984 4.27 KB (4,368 bytes)10/4/2000 12:27:42 PM Microsoft Corporationc:\winnt\system32\inetsrv\rpcref.dll
iisrtl.dll 5.00.0984 120.77 KB (123,664 bytes) 10/4/2000 12:27:42 PM Microsoft Corporationc:\winnt\system32\iisrtl.dll
inetinfo.exe 5.00.0984 14.27 KB (14,608 bytes) 10/4/2000 12:27:41 PM Microsoft Corporation c:\winnt\system32\inetsrv\inetinfo.exe
netui1.dll 5.00.2134.1 210.27 KB (215,312 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\netui1.dll
netui0.dll 5.00.2134.1 70.27 KB (71,952 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\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 Corporationc:\winnt\system32\wshnetbs.dll
rapilib.dll 5.00.2167.1 25.27 KB (25,872 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\rapilib.dll
rsvpsp.dll 5.00.2167.1 74.77 KB (76,560 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\rsvpsp.dll
ntmarta.dll5.00.2158.1 98.77 KB (101,136 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\ntmarta.dll
provthrd.dll 1.50.1085.0000 68.07 KB (69,708 bytes) 10/4/2000 4:32:07 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 Corporationc:\winnt\system32\wbem\ntevt.dll
perfos.dll 5.00.2155.1 21.27 KB (21,776 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\perfos.dll
psapi.dll 5.00.2134.1 28.27 KB (28,944 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\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 1.50.1085.0000 1.03 MB (1,077,306 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\cimwin32.dll
wbemsvc.dll 1.50.1085.0000 140.07 KB (143,430 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\wbemsvc.dll
wbemess.dll 1.50.1085.0001 352.05 KB (360,503 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\wbemess.dll
fastprox.dll1.50.1085.0001 144.08 KB (147,534 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\fastprox.dll
wbemcore.dll 1.50.1085.0001 632.05 KB (647,224 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\wbemcore.dll
wbemcomn.dll 1.50.1085.0001 684.05 KB (700,472 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\wbemcomn.dll
winmgmt.exe 1.50.1085.0001 188.05 KB (192,567 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\winmgmt.exe
simptcp.dll5.00.2134.1 19.27 KB (19,728 bytes) 10/4/2000 12:26:54 PM Microsoft Corporationc:\winnt\system32\simptcp.dll
tcpsvcs.exe5.00.2134.1 24.77 KB (25,360 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\tcpsvcs.exe
msidle.dll 5.00.2920.0000 6.27 KB (6,416 bytes)12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\msidle.dll
mstask.exe4.71.2137.1 115.27 KB (118,032 bytes) 10/4/2000 4:32:07 PM Microsoft Corporationc:\winnt\system32\mstask.exe
traffic.dll 5.00.2139.1 30.77 KB (31,504 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\traffic.dll
rsvp.exe 5.00.2167.1 172.77 KB (176,912 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\rsvp.exe
regsvc.exe 5.00.2155.1 65.27 KB (66,832 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\regsvc.exe
llsrpc.dll 5.00.2149.1 45.77 KB (46,864 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\llsrpc.dll
llssrv.exe 5.00.2167.1 114.27 KB (117,008 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\llssrv.exe
netshell.dll5.00.2176.1 456.77 KB (467,728 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\netshell.dll
netman.dll 5.00.2175.1 88.77 KB (90,896 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\netman.dll
ntmsdba.dll 5.00.2187.1 167.77 KB (171,792 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\ntmsdba.dll
rasdlg.dll 5.00.2194.1 514.27 KB (526,608 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\rasdlg.dll
netcfgx.dll 5.00.2175.1 533.77 KB (546,576 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\netcfgx.dll
rasmans.dll5.00.2188.1 146.77 KB (150,288 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\rasmans.dll
sens.dll 5.00.2163.1 36.77 KB (37,648 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\sens.dll
iashlpr.dll 5.00.2184.1 33.27 KB (34,064 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\iashlpr.dll
iasacct.dll 5.00.2134.1 28.27 KB (28,944 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\iasacct.dll
iasuserr.dll5.00.2134.1 25.77 KB (26,384 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\iasuserr.dll
iasnap.dll 5.00.2134.1 58.77 KB (60,176 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\iasnap.dll
iaspipe.dll 5.00.2134.1 41.77 KB (42,768 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\iaspipe.dll
expsrv.dll 6.0.8540 370.27 KB (379,152 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\expsrv.dll
vbajet32.dll 6.1.8268 30.27 KB (30,992 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\vbajet32.dll
msjtes40.dll 4.00.2927.8 232.27 KB (237,840 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\msjtes40.dll
oledb32r.dll 2.60.6526.0 68.27 KB (69,904 bytes) 10/5/2000 2:55:17 PM Microsoft Corporationc:\program files\common files\system\ole db\oledb32r.dll

comdlg32.dll 5.00.2920.0000 236.77 KB (242,448 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\comdlg32.dll
msdart.dll 2.60.6526.0 144.27 KB (147,728 bytes) 10/5/2000 2:55:16 PM Microsoft Corporationc:\winnt\system32\msdart.dll
oledb32.dll2.60.6526.0 448.27 KB (459,024 bytes) 10/5/2000 2:55:17 PM Microsoft Corporationc:\program files\common files\system\ole db\oledb32.dll
msjint40.dll 4.00.2927.2 148.27 KB (151,824 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\msjint40.dll
msjter40.dll 4.00.2927.2 52.27 KB (53,520 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\msjter40.dll
mswstr10.dll 4.00.2927.10 600.27 KB (614,672 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\mswstr10.dll
msjet40.dll4.00.2927.4 1.43 MB (1,495,312 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\msjet40.dll
msjetoledb40.dll 4.00.2927.2 340.27 KB (348,432 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\msjetoledb40.dll
iasrad.dll 5.00.2139.1 94.27 KB (96,528 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\iasrad.dll
iassam.dll 5.00.2160.1 96.27 KB (98,576 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\iassam.dll
iasads.dll 5.00.2134.1 73.77 KB (75,536 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\iasads.dll
ntmssvc.dll5.00.2187.1 390.77 KB (400,144 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\ntmssvc.dll
iaspolcy.dll5.00.2134.1 25.27 KB (25,872 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\iaspolcy.dll
iassvcs.dll 5.00.2160.1 58.77 KB (60,176 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\iassvcs.dll
iassdo.dll 5.00.2157.1 262.27 KB (268,560 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\iassdo.dll
ias.dll 5.00.2134.1 7.27 KB (7,440 bytes)12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\ias.dll
es.dll 1999.9.3422.21 231.77 KB (237,328 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\es.dll
mtxoci.dll 1999.9.3421.3 109.27 KB (111,888 bytes) 10/4/2000 12:27:01 PM Microsoft Corporationc:\winnt\system32\mtxoci.dll
resutils.dll 5.00.2191.1 39.77 KB (40,720 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\resutils.dll
clusapi.dll 5.00.2179.1 50.27 KB (51,472 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\clusapi.dll
msvcp50.dll 5.00.7051 552.50 KB (565,760 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\msvcp50.dll
xolehlp.dll 1999.9.3421.3 17.27 KB (17,680 bytes) 10/4/2000 12:27:00 PM Microsoft Corporationc:\winnt\system32\xolehlp.dll
msdtclog.dll 1999.9.3421.3 89.77 KB (91,920 bytes) 10/4/2000 12:27:00 PM Microsoft Corporation c:\winnt\system32\msdtclog.dll
mtxclu.dll 1999.9.3421.3 50.27 KB (51,472 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\mtxclu.dll
msdtcprx.dll 1999.9.3422.10 619.27 KB (634,128 bytes) 10/4/2000 12:27:01 PM Microsoft Corporation c:\winnt\system32\msdtcprx.dll
txfaux.dll 1999.9.3422.24 341.27 KB (349,456 bytes) 10/4/2000 12:27:00 PM Microsoft Corporationc:\winnt\system32\txfaux.dll
msdtctm.dll 1999.9.3422.12 1.02 MB (1,070,864 bytes) 10/4/2000 12:27:01 PM Microsoft Corporation c:\winnt\system32\msdtctm.dll
msdtc.exe 1999.9.3421.3 6.77 KB (6,928 bytes)10/4/2000 12:27:00 PM Microsoft Corporationc:\winnt\system32\msdtc.exe
wmi.dll 5.00.2191.1 6.27 KB (6,416 bytes)12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\wmi.dll
admwprox.dll 5.00.0984 31.77 KB (32,528 bytes) 10/4/2000 12:27:42 PM Microsoft Corporationc:\winnt\system32\admwprox.dll
inetpp.dll 5.00.2161.1 63.27 KB (64,784 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\inetpp.dll
win32spl.dll 5.00.2162.1 92.27 KB (94,480 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\win32spl.dll
usbmon.dll5.00.2165.1 11.27 KB (11,536 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\usbmon.dll
tcpmon.dll 5.00.2165.1 40.77 KB (41,744 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\tcpmon.dll
pjlmon.dll 5.00.2165.1 12.77 KB (13,072 bytes) 11/30/1999 6:39:36 PM Microsoft Corporationc:\winnt\system32\pjlmon.dll
cnbjmon.dll 5.00.2134.1 43.77 KB (44,816 bytes) 11/30/1999 6:38:48 PM Microsoft Corporation c:\winnt\system32\cnbjmon.dll
localspl.dll5.00.2191.1 244.77 KB (250,640 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\localspl.dll
spoolss.dll 5.00.2161.1 61.77 KB (63,248 bytes) 10/4/2000 12:10:53 PM Microsoft Corporationc:\winnt\system32\spoolss.dll
spoolsv.exe5.00.2161.1 43.77 KB (44,816 bytes) 10/4/2000 12:10:53 PM Microsoft Corporationc:\winnt\system32\spoolsv.exe
rpcss.dll 5.00.2181.1 229.27 KB (234,768 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\rpcss.dll
svchost.exe5.00.2134.1 7.77 KB (7,952 bytes)12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\svchost.exe
dssbase.dll5.00.2150.1 140.77 KB (144,144 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\dssbase.dll
oakley.dll 5.00.2174.1 420.27 KB (430,352 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\oakley.dll
mfc42u.dll6.00.8665.0 972.05 KB (995,384 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\mfc42u.dll
polagent.dll 5.00.2183.1 108.27 KB (110,864 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\polagent.dll
scecli.dll 5.00.2191.1 105.27 KB (107,792 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\scecli.dll
atl.dll 3.00.8449 57.56 KB (58,938 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\atl.dll
certcli.dll 5.00.2175.1 132.27 KB (135,440 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\certcli.dll
esent.dll 6.0.3939.6 1.07 MB (1,120,016 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\esent.dll
ntdsatq.dll 5.00.2181.1 31.27 KB (32,016 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\ntdsatq.dll
ntdsa.dll 5.00.2195.1 993.27 KB (1,017,104 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\ntdsa.dll
kdcsvc.dll 5.00.2181.1 133.77 KB (136,976 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\kdcsvc.dll
sfmapi.dll 5.00.2134.1 38.77 KB (39,696 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\sfmapi.dll
rassfm.dll 5.00.2168.1 21.27 KB (21,776 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\rassfm.dll
mpr.dll 5.00.2146.1 53.27 KB (54,544 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\mpr.dll
schannel.dll 5.00.2170.1 139.77 KB (143,120 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\schannel.dll
netlogon.dll 5.00.2182.1 347.77 KB (356,112 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\netlogon.dll
msv1_0.dll5.00.2164.1 94.77 KB (97,040 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\msv1_0.dll
kerberos.dll 5.00.2181.1 196.77 KB (201,488 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\kerberos.dll
msprivs.dll5.00.2154.1 41.50 KB (42,496 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\msprivs.dll
samsrv.dll 5.00.2192.1 357.77 KB (366,352 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\samsrv.dll
lsasrv.dll 5.00.2184.1 487.77 KB (499,472 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\lsasrv.dll
lsass.exe 5.00.2184.1 32.77 KB (33,552 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\lsass.exe

xactsrv.dll 5.00.2134.1 90.27 KB (92,432 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\xactsrv.dll
wmicore.dll 5.00.2178.1 70.77 KB (72,464 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\wmicore.dll
rasadhlp.dll 5.00.2168.1 7.27 KB (7,440 bytes)12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\rasadhlp.dll
winrnr.dll 5.00.2160.1 18.77 KB (19,216 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\winrnr.dll
rnr20.dll 5.00.2152.1 35.77 KB (36,624 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\rnr20.dll
wshtcpip.dll 5.00.2134.1 17.27 KB (17,680 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\wshtcpip.dll
msafd.dll 5.00.2153.1 54.27 KB (55,568 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\msafd.dll
mswsock.dll 5.00.2152.1 62.27 KB (63,760 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\mswsock.dll
msgsvc.dll 5.00.2181.1 33.77 KB (34,576 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\msgsvc.dll
browser.dll5.00.2142.1 48.27 KB (49,424 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\browser.dll
alrsvc.dll 5.00.2134.1 17.77 KB (18,192 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\alrsvc.dll
trkwks.dll 5.00.2166.1 88.77 KB (90,896 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\trkwks.dll
seclogon.dll 5.00.2135.1 15.77 KB (16,144 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\seclogon.dll
psbase.dll 5.00.2146.1 111.77 KB (114,448 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\psbase.dll
cryptsvc.dll 5.00.2181.1 61.77 KB (63,248 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\cryptsvc.dll
cryptdll.dll5.00.2135.1 41.27 KB (42,256 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\cryptdll.dll
wkssvc.dll 5.00.2181.1 95.27 KB (97,552 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\wkssvc.dll
srvsvc.dll 5.00.2178.1 79.27 KB (81,168 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\srvsvc.dll
cfgmgr32.dll 5.00.2134.1 16.77 KB (17,168 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\cfgmgr32.dll
dmserver.dll 2191.1.296.2 11.77 KB (12,048 bytes) 12/7/1999 7:00:00 AM VERITAS Software Corp. c:\winnt\system32\dmserver.dll
winsta.dll 5.00.2134.1 36.27 KB (37,136 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\winsta.dll
lmhsvc.dll 5.00.2134.1 9.27 KB (9,488 bytes)12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\lmhsvc.dll
dnsrslvr.dll5.00.2181.1 88.27 KB (90,384 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\dnsrslvr.dll
tapi32.dll 5.00.2182.1 123.27 KB (126,224 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\tapi32.dll
rasman.dll 5.00.2188.1 54.77 KB (56,080 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\rasman.dll
rasapi32.dll 5.00.2188.1 189.77 KB (194,320 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\rasapi32.dll
rtutils.dll 5.00.2168.1 43.77 KB (44,816 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\rtutils.dll
adsldpc.dll 5.00.2172.1 127.77 KB (130,832 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\adsldpc.dll
activeds.dll5.00.2172.1 172.77 KB (176,912 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\activeds.dll
mprapi.dll 5.00.2181.1 79.27 KB (81,168 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\mprapi.dll
iphlpapi.dll5.00.2173.2 67.77 KB (69,392 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\iphlpapi.dll
icmp.dll 5.00.2134.1 7.27 KB (7,440 bytes)12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\icmp.dll
dhcpcsvc.dll 5.00.2153.1 88.77 KB (90,896 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\dhcpcsvc.dll
eventlog.dll 5.00.2178.1 43.77 KB (44,816 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\eventlog.dll
ntdsapi.dll 5.00.2160.1 56.27 KB (57,616 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\ntdsapi.dll
scesrv.dll 5.00.2188.1 225.77 KB (231,184 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\scesrv.dll
umpnpmgr.dll 5.00.2182.1 86.27 KB (88,336 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\umpnpmgr.dll
services.exe 5.00.2134.1 86.77 KB (88,848 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\services.exe
clbcatq.dll 1999.9.3422.14 479.27 KB (490,768 bytes) 10/4/2000 12:26:54 PM Microsoft Corporationc:\winnt\system32\clbcatq.dll
oleaut32.dll 2.40.4512 600.27 KB (614,672 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\oleaut32.dll
cscui.dll 5.00.2172.1 227.27 KB (232,720 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\cscui.dll
winspool.drv 5.00.2167.1 109.77 KB (112,400 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\winspool.drv
winscard.dll 5.00.2134.1 77.27 KB (79,120 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\winscard.dll
wlnotify.dll 5.00.2164.1 53.27 KB (54,544 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\wlnotify.dll
cscdll.dll 5.00.2189.1 98.27 KB (100,624 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\cscdll.dll
lz32.dll 5.00.2134.1 9.77 KB (10,000 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\lz32.dll
version.dll 5.00.2134.1 15.77 KB (16,144 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\version.dll
rsabase.dll 5.00.2150.1 128.77 KB (131,856 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\rsabase.dll
mscat32.dll 5.131.2134.1 7.77 KB (7,952 bytes)12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\mscat32.dll
ole32.dll 5.00.2181.1 966.27 KB (989,456 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\ole32.dll
imagehlp.dll 5.00.2195.1 125.27 KB (128,272 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\imagehlp.dll
msasn1.dll 5.00.2134.1 51.27 KB (52,496 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\msasn1.dll
crypt32.dll 5.131.2173.1 465.77 KB (476,944 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\crypt32.dll
wintrust.dll5.131.2143.1 162.27 KB (166,160 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\wintrust.dll
setupapi.dll 5.00.2183.1 554.27 KB (567,568 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\setupapi.dll
winmm.dll 5.00.2161.1 184.77 KB (189,200 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\winmm.dll
comctl32.dll 5.81 540.27 KB (553,232 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\comctl32.dll
shlwapi.dll5.00.2920.0000 282.77 KB (289,552 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\shlwapi.dll
shell32.dll 5.00.2920.0000 2.24 MB (2,352,400 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\shell32.dll
msgina.dll 5.00.2191.1 309.77 KB (317,200 bytes) 12/7/1999 7:00:00 AM Microsoft Corporationc:\winnt\system32\msgina.dll
wsock32.dll 5.00.2152.1 21.27 KB (21,776 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\wsock32.dll

dnsapi.dll 5.00.2181.1 129.77 KB (132,880 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\dnsapi.dll
wldap32.dll 5.00.2168.1 155.77 KB (159,504 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\wldap32.dll
ws2help.dll 5.00.2134.1 17.77 KB (18,192 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\ws2help.dll
ws2_32.dll 5.00.2134.1 69.77 KB (71,440 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\ws2_32.dll
samlib.dll 5.00.2160.1 46.27 KB (47,376 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\samlib.dll
netrap.dll 5.00.2134.1 11.27 KB (11,536 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\netrap.dll
netapi32.dll 5.00.2194.1 302.77 KB (310,032 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\netapi32.dll
profmap.dll 5.00.2181.1 29.27 KB (29,968 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\profmap.dll
secur32.dll 5.00.2154.1 46.77 KB (47,888 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\secur32.dll
sfc.dll 5.00.2164.1 84.27 KB (86,288 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\sfc.dll
nddeapi.dll 5.00.2137.1 15.27 KB (15,632 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\nddeapi.dll
userenv.dll 5.00.2185.1 361.27 KB (369,936 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\userenv.dll
user32.dll 5.00.2180.1 393.27 KB (402,704 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\user32.dll
gdi32.dll 5.00.2180.1 228.77 KB (234,256 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\gdi32.dll
rpcrt4.dll 5.00.2193.1 434.27 KB (444,688 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\rpcrt4.dll
advapi32.dll 5.00.2191.1 349.27 KB (357,648 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\advapi32.dll
kernel32.dll 5.00.2191.1 715.27 KB (732,432 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\kernel32.dll
msvcrt.dll 6.10.8637.0 288.09 KB (295,000 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\msvcrt.dll
winlogon.exe 5.00.2182.1 173.27 KB (177,424 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\winlogon.exe
sfcfiles.dll 5.00.2195.1 973.27 KB (996,624 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\sfcfiles.dll
ntdll.dll 5.00.2163.1 469.77 KB (481,040 bytes) 12/7/1999 7:00:00 AM Microsoft Corporation c:\winnt\system32\ntdll.dll
smss.exe 5.00.2170.1 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
Alerter Alerter 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\dfssvc.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\inetsrv\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 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 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 mnmsrvc Stopped Manual Own Process c:\winnt\system32\mnmsrvc.exe Normal LocalSystem 0
Distributed Transaction Coordinator MSDTC Running Auto Own Process c:\winnt\system32\msdtc.exe Normal LocalSystem 0
Windows Installer MSIServer Stopped Manual Share Process c:\winnt\system32\msiexec.exe /v Normal LocalSystem 0
Network DDE NetDDE Stopped Manual Share Process c:\winnt\system32\netdde.exe Normal LocalSystem 0
Network DDE DSDM NetDDEdsdm Stopped Manual Share Process c:\winnt\system32\netdde.exe Normal LocalSystem 0
Net Logon Netlogon Stopped Manual Share Process c:\winnt\system32\lsass.exe Normal LocalSystem 0
Network Connections Netman Running Manual Share Process c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
File Replication NtFrs Stopped Manual Own Process c:\winnt\system32\ntfrs.exe Ignore LocalSystem 0
NT LM Security Support Provider NtLmSsp Stopped Manual Share Process c:\winnt\system32\lsass.exe Normal LocalSystem 0
Removable Storage NtmsSvc Running Auto Share Process c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Plug and Play PlugPlay Running Auto Share Process c:\winnt\system32\services.exe Normal LocalSystem 0
IPSEC Policy Agent PolicyAgent Running Auto Share Process c:\winnt\system32\lsass.exe Normal LocalSystem 0
Protected Storage ProtectedStorage Running Auto Share Process c:\winnt\system32\services.exe Normal LocalSystem 0
Remote Access Auto Connection Manager RasAuto Stopped Manual Share Process c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Remote Access Connection Manager RasMan Stopped Manual Share Process c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0

Routing and Remote Access RemoteAccess Stopped Disabled
Share Process c:\winnt\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Remote Registry Service RemoteRegistry Running Auto
Own Process c:\winnt\system32\regsvc.exe Normal LocalSystem
0
Remote Procedure Call (RPC) Locator RpcLocator Stopped
Manual Own Process c:\winnt\system32\locator.exe Normal
LocalSystem 0
Remote Procedure Call (RPC) RpcSs Running Auto Share
Process c:\winnt\system32\svchost -k rpcss Normal LocalSystem 0
QoS Admission Control (RSVP) RSVP Running Auto Own Process
c:\winnt\system32\rsvp.exe -s Normal LocalSystem 0
Security Accounts Manager SamSs Running Auto Share
Process c:\winnt\system32\lsass.exe Normal LocalSystem 0
Smart Card Helper SCardDrv Stopped Manual Share Process
c:\winnt\system32\scardsvr.exe Ignore LocalSystem 0
Smart Card SCardSvr Stopped Manual Share Process
c:\winnt\system32\scardsvr.exe Ignore LocalSystem 0
Task Scheduler Schedule Running Auto Share Process
c:\winnt\system32\mstask.exe Normal LocalSystem 0
RunAs Service seclogon Running Auto Share Process
c:\winnt\system32\services.exe Ignore LocalSystem 0
System Event Notification SENS Running Auto Share
Process c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem
0
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 Share
Process c:\winnt\system32\tcpsvcs.exe Normal LocalSystem 0
Print Spooler Spooler Running Auto Own Process
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 Own Process
c:\winnt\system32\termsrv.exe Normal LocalSystem 0
Telnet TlntSvr Stopped Manual Own Process
c:\winnt\system32\tlntsvr.exe Normal LocalSystem 0
Distributed Link Tracking Server TrkSvr Stopped Manual Share
Process c:\winnt\system32\services.exe Normal LocalSystem 0
Distributed Link Tracking Client TrkWks Running Auto Share
Process c:\winnt\system32\services.exe Normal LocalSystem 0
Uninterruptible Power Supply UPS Stopped Manual Own Process
c:\winnt\system32\ups.exe Normal LocalSystem 0
Utility Manager UtilMan Stopped Manual Own Process
c:\winnt\system32\utilman.exe Normal LocalSystem 0
Windows Time W32Time Stopped Manual Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
World Wide Web Publishing Service W3SVC Running Auto
Share Process c:\winnt\system32\inetsrv\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 SQLCLIENT10\Administrator:Accessories
SQLCLIENT10\Administrator
Accessories\Accessibility
SQLCLIENT10\Administrator:Accessories\Accessibility
SQLCLIENT10\Administrator
Accessories\Entertainment
SQLCLIENT10\Administrator:Accessories\Entertainment
SQLCLIENT10\Administrator
Accessories\System Tools
SQLCLIENT10\Administrator:Accessories\System Tools
SQLCLIENT10\Administrator
Administrative Tools SQLCLIENT10\Administrator:Administrative Tools
SQLCLIENT10\Administrator
Startup SQLCLIENT10\Administrator:Startup
SQLCLIENT10\Administrator

[Startup Programs]

Program Command User NameLocation
Ibmmon.exe ibmmon.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 mspaint.exe

[Internet Explorer 5]

[Following are sub-categories of this main category]

[Summary]

Item Value
Version 5.00.2920.0000
Build 52920
Product ID 51876-270-6758736-05321
Application Path C:\Program Files\Internet Explorer
Language English (United States)
Active Printer Not Available

Cipher Strength 56-bit

Content Advisor Disabled
IEAK Install No

[File Versions]

File Version Size Date Path Company
advapi32.dll 5.0.2191.1 349 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
advpack.dll 5.0.2920.0 87 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
browselc.dll 5.0.2920.0 35 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
browseui.dll 5.0.2920.0 793 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
ckcnv.exe 5.0.2189.1 9 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
comctl32.dll 5.81.2920.0 540 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
crypt32.dll 5.131.2173.1 466 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
enhsig.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 5.0.2920.0 57 KB 12/7/1999 7:00:00 AM
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 5.0.2195.1 125 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
imghelp.dll<File Missing> Not Available Not Available
Not Available Not Available
inseng.dll 5.0.2920.0 72 KB 12/7/1999 7:00:00 AM
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
jscript.dll 5.1.0.4615 476 KB 12/7/1999 7:00:00 AM
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 5.0.2920.0 2302 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
msjava.dll 5.0.3234.0 918 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
msoss.dll <File Missing> Not Available Not Available
Not Available Not Available
msxml.dll 5.0.2920.0 521 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
occache.dll 5.0.2920.0 86 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
ole32.dll 5.0.2181.1 966 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
oleaut32.dll 2.40.4512.1 600 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
olepro32.dll 5.0.4512.1 160 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
rsabase.dll 5.0.2150.1 129 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
rsaenh.dll <File Missing> Not Available Not Available
Not Available Not Available
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 5.0.2170.0 140 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
shdoc401.dll <File Missing> Not Available Not
Available Not Available Not Available
shdocvw.dll 5.0.2920.0 1078 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
shell32.dll 5.0.2920.0 2297 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
shlwapi.dll 5.0.2920.0 283 KB 12/7/1999 7:00:00 AM
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 5.0.2920.0 427 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
vbscript.dll 5.1.0.4615 428 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
webcheck.dll 5.0.2920.0 252 KB 12/7/1999 7:00:00 AM
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 5.0.2920.0 457 KB 12/7/1999 7:00:00 AM
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 5.131.2143.1 162 KB 12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
wsock.vxd <File Missing> Not Available Not Available
Not Available Not Available
wsock32.dll 5.0.2152.1 21 KB 12/7/1999 7:00:00 AM
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 7052 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 10/4/2000 to 9/10/2100 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

TPCC.AllTxns:
Activation:
Enable Object Pooling selected
Minimum Pool Size: 37
Maximum Pool Size: 37
Creating Timeout: 60,000
Enable Object Construction
Enable Just in Time Activation
Concurrency:
Concurrency Required

TPCC Application Registry Parameters

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPCC]
"Path"="c:\\inetpub\\wwwroot\\"
"NumberOfDeliveryThreads"=dword:00000004
"MaxConnections"=dword:00001b58
"MaxPendingDeliveries"=dword:000003e8
"DB_Protocol"="DBLIB"
"TxnMonitor"="COM"
"DbServer"="ibmserv3"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"=""
"COM_SinglePool"="YES"

Microsoft Internet Information Service Registry Parameters

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,00,00
"PoolThreadLimit"=dword:000000be
"ThreadTimeout"=dword:00015180

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000842
"Last Help"=dword:00000843
"First Counter"=dword:00000802

Worldwide Web Service Registry Parameters

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,00,3a,00,5c,00,57,00,49,00,4e,00,4e,00,54,00,5c,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\\iiscrmap.dll"
"AccessDeniedMessage"="Error: Access is Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\\WINNT\\System32\\LogFiles"
"AcceptExOutstanding"=dword:00000028

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\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\\inetsrv\\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\Performance]
"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\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\

20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\

00,05,20,00,00,00,23,02,00,00,72,00,73,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Enum]
"0"="Root\\LEGACY_W3SVC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

RTE Input Parameters

Profile: 2750wh-28segments
File Path: C:\benchcrf\2750wh-28segments.pro
Version: 1.0.1

Number of Engines: 28

Name: DRIVER1
Description: rte11_toclient
Directory: c:\benchcrf\logs\driver1.log
Machine: rte10
Parameter Set: PARAM2
Index: 0
Seed: 28630
Configured Users: 990
Pipe Name: DRIVER11594390
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER10
Description: rte23_toclient
Directory: c:\benchcrf\logs\driver10.log
Machine: rte20
Parameter Set: PARAM2
Index: 900000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER101942843
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER11
Description: rte24_toclient
Directory: c:\benchcrf\logs\river11.log
Machine: rte20
Parameter Set: PARAM2
Index: 1000000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER111969046
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER12
Description: rte25_toclient
Directory: c:\benchcrf\logs\driver12.log
Machine: rte20
Parameter Set: PARAM2
Index: 1100000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER121997234
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER13
Description: rte26_toclient
Directory: c:\benchcrf\logs\driver13.log
Machine: rte20
Parameter Set: PARAM2
Index: 1200000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER132028921
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER14
Description: rte27_toclient
Directory: c:\benchcrf\logs\driver14.log
Machine: rte20
Parameter Set: PARAM2
Index: 1300000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER142053484
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER15
Description: rte31_toclient
Directory: c:\benchcrf\logs\driver15.log
Machine: rte30
Parameter Set: PARAM2
Index: 1400000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER152082187
Connect Rate: 500
Start Rate: 0

CLIENT_NURAND: 208
CPU: 0

Name: DRIVER16
Description: rte32_toclient
Directory: c:\benchcrf\logs\driver16.log
Machine: rte30
Parameter Set: PARAM2
Index: 1500000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER162114078
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER17
Description: rte33_toclient
Directory: c:\benchcrf\logs\driver17.log
Machine: rte30
Parameter Set: PARAM2
Index: 1600000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER172146265
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER18
Description: rte34_toclient
Directory: c:\benchcrf\logs\driver18.log
Machine: rte30
Parameter Set: PARAM2
Index: 1700000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER182173968
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER19
Description: rte35_toclient
Directory: c:\benchcrf\logs\driver19.log
Machine: rte30
Parameter Set: PARAM2
Index: 1800000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER192201718
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER2
Description: rte12_toclient
Directory: c:\benchcrf\logs\driver2.log
Machine: rte10
Parameter Set: PARAM2
Index: 100000000
Seed: 28630
Configured Users: 990
Pipe Name: DRIVER21672640
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER20
Description: rte36_toclient
Directory: c:\benchcrf\logs\driver20.log
Machine: rte30
Parameter Set: PARAM2
Index: 1900000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER202229015
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER21
Description: rte37_toclient
Directory: c:\benchcrf\logs\driver21.log
Machine: rte30
Parameter Set: PARAM2
Index: 2000000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER212261093
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER22
Description: rte41_toclient
Directory: c:\benchcrf\logs\driver22.log
Machine: rte40
Parameter Set: PARAM2
Index: 2100000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER222287671
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER23
Description: rte42_toclient
Directory: c:\benchcrf\logs\driver23.log
Machine: rte40
Parameter Set: PARAM2
Index: 80000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER232312734
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER24
Description: rte43_toclient
Directory: c:\benchcrf\logs\driver24.log
Machine: rte40
Parameter Set: PARAM2
Index: 70000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER242405515
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER25
Description: rte44_toclient
Directory: c:\benchcrf\logs\driver25.log
Machine: rte40
Parameter Set: PARAM2
Index: 60000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER252459453
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER26
Description: rte45_toclient
Directory: c:\benchcrf\logs\driver26.log
Machine: rte40
Parameter Set: PARAM2
Index: 50000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER262492625
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER27
Description: rte46_toclient
Directory: c:\benchcrf\logs\driver27.log
Machine: rte40
Parameter Set: PARAM2
Index: 40000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER272532578
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER28
Description: rte47_toclient
Directory: c:\benchcrf\logs\driver28.log
Machine: rte40
Parameter Set: PARAM2
Index: 30000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER282567687
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER3
Description: rte13_toclient
Directory: c:\benchcrf\logs\driver3.log
Machine: rte10
Parameter Set: PARAM2
Index: 200000000
Seed: 28630
Configured Users: 990
Pipe Name: DRIVER31704859
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER4
Description: rte14_toclient
Directory: c:\benchcrf\logs\driver4.log
Machine: rte10
Parameter Set: PARAM2
Index: 300000000
Seed: 28630
Configured Users: 990
Pipe Name: DRIVER41739734
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER5
Description: rte15_toclient
Directory: c:\benchcrf\logs\driver5.log
Machine: rte10
Parameter Set: PARAM2
Index: 400000000
Seed: 28630
Configured Users: 990
Pipe Name: DRIVER51768187
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER6
Description: rte16_toclient
Directory: c:\benchcrf\logs\driver6.log
Machine: rte10
Parameter Set: PARAM2
Index: 500000000
Seed: 28630
Configured Users: 990
Pipe Name: DRIVER61803546
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER7
Description: rte17_toclient
Directory: c:\benchcrf\logs\driver7.log
Machine: rte10
Parameter Set: PARAM2
Index: 600000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER71838546
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER8
Description: rte21_toclient
Directory: c:\benchcrf\logs\driver8.log
Machine: rte20
Parameter Set: PARAM2
Index: 700000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER81865281
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER9

Description: rte22_toclient
Directory: c:\benchcrf\logs\driver9.log
Machine: rte20
Parameter Set: PARAM2
Index: 800000000
Seed: 28630
Configured Users: 980
Pipe Name: DRIVER91899625
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Number of User groups: 28

Driver Engine: DRIVER1
IIS Server: client10_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1 - 99
w_id Max Warehouse: 2750
Scale: Normal
User Count: 990
District id: 1
Scale Down: No

Driver Engine: DRIVER9
IIS Server: client20_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 791 - 888
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER11
IIS Server: client20_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 987 - 1084
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER21
IIS Server: client30_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1967 - 2064
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER12
IIS Server: client20_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1085 - 1182
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER13
IIS Server: client20_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1183 - 1280
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER15
IIS Server: client30_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1379 - 1476
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER25
IIS Server: client40_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2359 - 2456
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER16
IIS Server: client30_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1477 - 1574
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER17
IIS Server: client30_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1575 - 1672
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER19
IIS Server: client30_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1771 - 1868
w_id Max Warehouse: 2750

Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER28
IIS Server: client40_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2653 - 2750
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER2
IIS Server: client10_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 100 - 198
w_id Max Warehouse: 2750
Scale: Normal
User Count: 990
District id: 1
Scale Down: No

Driver Engine: DRIVER20
IIS Server: client30_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1869 - 1966
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER22
IIS Server: client40_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2065 - 2162
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER6
IIS Server: client10_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 496 - 594
w_id Max Warehouse: 2750
Scale: Normal
User Count: 990
District id: 1
Scale Down: No

Driver Engine: DRIVER23
IIS Server: client40_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2163 - 2260
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER24
IIS Server: client40_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2261 - 2358
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER26
IIS Server: client40_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2457 - 2554
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER10
IIS Server: client20_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 889 - 986
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER27
IIS Server: client40_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2555 - 2652
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER3
IIS Server: client10_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 199 - 297
w_id Max Warehouse: 2750
Scale: Normal
User Count: 990
District id: 1
Scale Down: No

Driver Engine: DRIVER14
IIS Server: client20_torte
SQL Server: ibmserv3
User: sa
Protocol: Html

w_id Range: 1281 - 1378
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER4
IIS Server: client10_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 298 - 396
w_id Max Warehouse: 2750
Scale: Normal
User Count: 990
District id: 1
Scale Down: No

Driver Engine: DRIVER5
IIS Server: client10_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 397 - 495
w_id Max Warehouse: 2750
Scale: Normal
User Count: 990
District id: 1
Scale Down: No

Driver Engine: DRIVER7
IIS Server: client10_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 595 - 692
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER18
IIS Server: client30_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1673 - 1770
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Driver Engine: DRIVER8
IIS Server: client20_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 693 - 790
w_id Max Warehouse: 2750
Scale: Normal
User Count: 980
District id: 1
Scale Down: No

Number of Parameter Sets: 2

~Default

	5.00	0.10
	5.00	0.10
	5.00	0.10
	20.00	0.10
	5.00	0.10
	5.00	0.10
	5.00	0.10
	20.00	0.10
	5.00	0.10

Default Parameter Set

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	10.00	12.05		18.01		0.10
Payment	10.00	12.05		3.01		0.10
Delivery	1.00	5.05		2.01		0.10
Stock Level	1.00	5.05		2.01		0.10
Order Status	1.00	10.05		2.01		0.10

PARAM2
first try

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	44.84	12.05		18.01		0.10
Payment	43.04	12.05		3.01		0.10
Delivery	4.04	5.05		2.01		0.10
Stock Level	4.04	5.05		2.01		0.10
Order Status	4.04	10.05		2.01		0.10

Appendix D: 60-Day Space

TPC-C 60-Day Space Requirements						
Warehouses	2,750				tpmC	34264.90
Table	Rows	Data KB	Index KB	Extra 5% KB	8HR Space	Total Space KB
Warehouse	2,750	296	40	16.80		352.80
District	27,500	3,056	40	154.80		3,250.80
Item	100,000	9,528	56	479.20		10,063.20
New-Order	24,750,000	391,312	920		220,000.00	612,232.00
History	82,500,000	4,583,344	64		913,745.55	5,497,153.55
Orders	82,500,000	2,528,736	1,149,928		733,376.31	4,412,040.31
Customer	82,500,000	60,000,000	3,577,784	3,178,889.20		66,756,673.20
Order-Line	824,994,449	51,562,160	109,152		10,301,162.70	61,972,474.70
Stock	275,000,000	88,000,000	164,520	4,408,226.00		92,572,746.00
Totals		207,078,432	5,002,504	7,587,766.00	12,168,284.56	231,836,986.56
Segment	LogDev Cnt.	Segment Size	Needed	Overhead		Not Needed
misc	4	74,547,200	72,507,567	725,076		1,314,556.96
cs	4	160,972,800	159,329,419	1,593,294		50,086.61
master, msdb,model	1	13,312	13,312			
tpcc_root	1	8,192	8,192			
tempdb	1	8,704	8,704			
Totals		235,550,208	231,867,194.56	2,318,369.87		1,364,643.57
Dynamic Space	58,674,240.00	Sum of Data for Order, Order-Line and History				
Static Space	163,312,831.87	Data + Index + 5% Space + Overhead - Dynamic Space				
Free Space	12,198,492.56	Total Segment Size - Dynamic Space - Static Space - Not needed				
Daily Growth	11,697,262.35	(Dynamic Space/W * 62.5)* tpmC				
Daily Spread	(5,347,400.96)	Free Space - 1.5 * Daily Growth (Zero If Negative)				
60-Day Space (KB)	865,148,572.79	Static Space + 180 (Daily Growth + Daily Spread)				
60-Day Space (GB)	825.07	60-Day Space in GB (Excludes OS,Paging and RDBMS Logs)				
Available (GB)	2,367.68					
Log file storage requirement						
Log size (MB)	85,000.00	Total Size of Log File				
% log used	11.70	% of Log File Used During Entire Run				
Total N-O Txn	2,044,848.00	Total Count of New-Order Transactions during Entire Run				
Log / N-O Txn	4.98	KB of Log per New-Order Transaction				
8 Hour log (GB)	78.10	8 Hours of Log in GB (Excluding Space for Redundancy)				
Log configured (GB)	101.58					
Disk Capacity	MB	GB				
9.1GB	8,678	8.45				
18.2GB	17,356	16.93				
Space Usage	GB Needed	Disks Measured	Disks Priced	Disk Size	GB Priced	GB Usable
60-Day Space DB	825.07	168	168	9.1GB	1,419.60	1,419.6
		56	56	18.2GB	948.08	948.08
Extra Disks		0	0		0	0
Total DB		224	224		2,367.68	2,367.68
8hr Log + Mirror	78.1	12	12	18.2GB	203.16	101.58
OS, SQL Server	4.00	1	1	9.1GB	8.45	8.45
Total Space	907.17	237	237		2,579.29	2,477.71

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

April 3, 2001

IBM Corp.
Chris King
IBM Corporation
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 V5.0 benchmark testing.

All pricing shown is in US Dollars ($).

Part Number	Description	Unit Price	Quantity	Price
810-00846	**SQL Server 2000 Enterprise Edition** *Per processor licensing* *Discount schedule: Open Program Level C*	$ 16,541	4	$ 66,164
C11-00821	**Windows 2000 Server** *Server license only - No CALs* *Discount schedule: Open Program - No Level*	$ 738	1	$ 738
C10-00475	**Windows 2000 Advanced Server** *Server license only - No CALs* *Discount schedule: Open Program - No Level*	$ 2,399	1	$ 2,399
048-00317	**Visual C++ Professional 6.0 Win32**	$ 549	1	$ 549
	3-year maintenance for above software	$ 2,095	1	$ 6,285

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: Pbwyl0103049720
Please include this Reference ID in any correspondence regarding this price quote.

Mylex/IBM Extreme RAID 2000 Quotation

Ms. Chris King, April 5, 2001
IBM Netfinity Performance Group

Dear Ms. King,

Mylex is pleased to submit the following quotation for ExtremeRAID 2000 controller.

Mylex P/N :	**Description**
E2000-4-32NB	(4 external +2 internal chnl, 32MB cache, no BBU)

Suggested Retail Price: $1,872.00 ea

Notes: Above price is based on FOB, ex factory, Fremont, California and firm for 90 days.

Lead time: 45 Days ARO

Product is covered by 5 year warranty.

Failed product will be repaired or replaced within 7 days.

Regards,

Robert Kelly - Director, Strategic Sales

Cc: Steve Page - Director PCI Marketing

Software House International
2880 Zanker Blvd. #103
San Jose, CA 95134
Matthew Martin
National Account Executive
1-800-766-6357

Description	Part Number	Unit Price	Qty	Extended Price
8-Port 10/100Mbps Hub Nway Fast Ethernet Switch	NX-DSS8	25	7	$175

Prices are valid for 90 days from March 16, 2001.