TPC Benchmark® C
Full Disclosure Report

hp server rx5670

using Microsoft SQL Server 2000 Enterprise Edition 64-bit
on Microsoft Windows .NET Enterprise Server 2003

First Edition
December 16, 2002

First Edition - December 16, 2002

Printed in U.S.A., December 16, 2002

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark® C test conducted on the hp server rx5670 in a client/server configuration, using Microsoft SQL Server 2000 Enterprise Edition 64-bit and Microsoft COM+ Transaction Monitor. The operating system used for the benchmark was Microsoft Windows .NET Enterprise Server 2003.

TPC Benchmark® C Metrics

The standard TPC Benchmark ® C metrics, tpmC® (transactions per minute), price per tpmC ® (three year capital cost per measured tpmC®), and the availability date are reported as required by the benchmark specification.

Standard and Executive Summary Statements

The following pages contain the executive summary of the benchmark results for the hp server rx5670 system. The Standard System Summary is given below.

Company Name	System Name	Database Software	Operating System
Hewlett-Packard Company	hp server rx5670	Microsoft SQL Server 2000 Enterprise Edition 64-bit	Microsoft Windows .NET Enterprise Server 2003
Total System Cost	**TPC-C Throughput**	**Price/Performance**	**Availability Date**
$441,021.80	87741.45 tpmC	$5.03 per tpmC	February 12, 2003

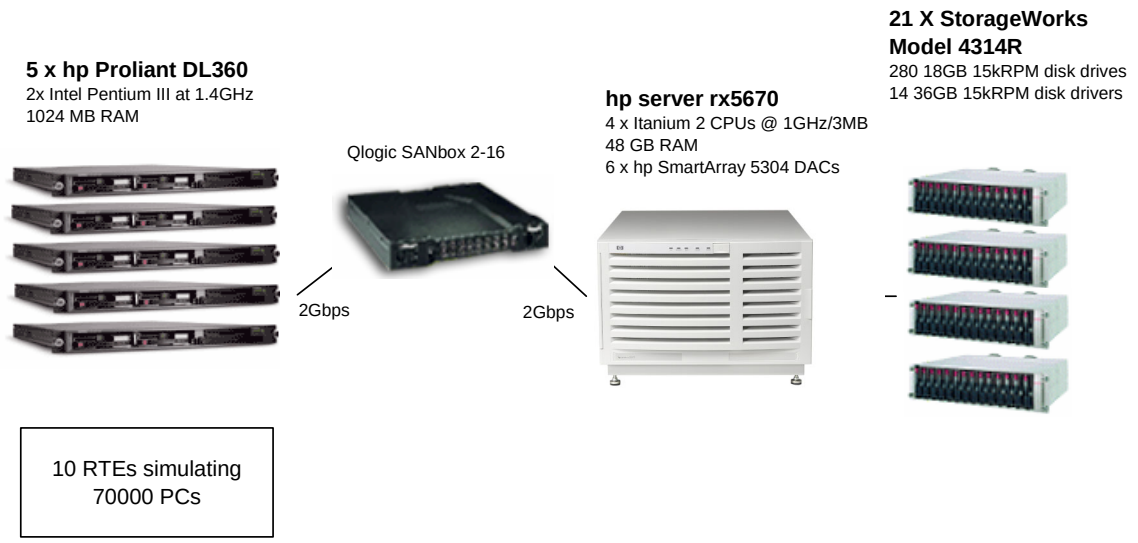
Auditor

The benchmark configuration, environment and methodology used to produce and validate the test results, and the pricing model used to calculate the cost per tpmC® , were audited by Francois Raab of Infosizing to verify compliance with the relevant TPC specifications.

hp server rx5670 Client/Server

TPC-C Revision 5.0

Total System Cost	TPC Throughput	Price/Performance	Report Date	Availability Date
$441,022	87741.45 tpmC	$5.03 per tpmC	December 16, 2002	February 12, 2003
Processors	**Database Manager**	**Operating System**	**Other Software**	**Number of Users**
4 Intel Itanium 2 CPUs at 1 GHz 3 MB iL3 Cache	Microsoft SQL Server 2000 Enterprise Edition 64-bit	Microsoft Windows .NET Enterprise Server 2003	Microsoft Visual C++ Microsoft COM+ Transaction Monitor	70,000



System Components	Server		Each Client	
	Qty	Type	Qty	Type
Processors	4	Intel Itanium 2 CPUs at 1 GHz	2	1.4GHz Pentium III
Cache Memory		3 MB iL3 cache		512kbyte L2 Cache
Memory	48	1 Gbyte	4	256 Mbyte
Disk Controllers	6	HP SmartArray 5304	1	HP SCSI-2 Controller
Disk Drives	280	HP Hot-swap 18GB 15KRPM SCSI	1	18 Gbyte disk
	1	HP Hot-swap 36GB U320		
Total Storage		5002.4 Gbyte		16.8 Gbyte
Tape Drives	1	HP Surestore DAT40e		
Terminals	1	Console Terminal	1	Console Terminal

Hewlett Packard Company	hp server rx5670 client/server			TPC-C Rev. 5.0 Report Date: 16 December, 2002		
Description	Price Key	Part Numbr	Unit Price	Qty	Extended Price	3 Yr Maint Price
1GHz Itanium 2 w/ 3MB iL3 cache, 0 MB RAM, 0 disk	1	A6838A	$26,494	1	$26,494	
CPU upgrade Itanium 2, 1GHz w/3MB iL3 cache	1	A6836A	$8,250	3	$24,750	
4GB PC2100 DDR-SDRAM (4x1GB DIMMs)	1	A6834A	$8,000	12	$96,000	
Memory Carrier Board	1	A6747A	$1,981	2	$3,962	
HP 36GB, 15krpm Ultra320 hot-swap disk	1	A7049A	$996	2	$1,992	
HP Rackmount Kit Factory	1	A5580A	$134	1	$134	
DVD Rom drive	1	A5557B	$450	1	$450	
Graphics USB Card	1	A6869A	$349	1	$349	
HP USB keyboard and mouse	1	A7861A	$32	1	$32	
HP Smart Array Controller 5304	2	*283551-B21*	$2,247	6	$13,482	
Qlogic QLA-2350 Fibre-Channel VI Adapter	3	QLA2350-BK	$2,095	1	$2,095	
5m LC to LC Cable Kit	2	221692-B22	$82	1	$82	
2GB SFP Adapter Kit	2	221470-B21	$369	1	$369	
S5500 15 carbon / silver monitor	2	261602-001	$139	1	$139	
HP Rack Model 9142 (42U - Opal) - Flat Pallet	2	120663-B21	$1,352	2	$2,704	
HP Power Distribution Unit 120-240V	1	E7671A	$146	3	$438	
UPS R1500 XR	2	204404-001	$886	1	$886	
HP Hardware Support 3 yr, 24x7, 4 hr rx5670	1	H4405Y#6BO	$7,052	1		$7,052
HP Hardware Support 3 yr, 24x7, 4 hr addt'l CPU	1	H4405Y#6BP	$1,153	3		$3,459
20/40 GB DAT Drive, External	1	C5687B	$1,300	1	$1,300	
Storageworks enclosure 4314R	2	190209-001	$2,955	20	$59,100	
18GB, 15krpm Ultra3 Wide disk	2	188122-B22	$390	280	$109,200	
18GB, 15krpm Ultra3 Wide disk (10% spares)	2	188122-B22	$390	28	$10,920	
Storageworks enclosure 4354R	2	190211-001	$3,523	1	$3,523	
36GB, 15krpm Ultra3 Wide disk	2	232916-B22	$605	14	$8,470	
36GB, 15krpm Ultra3 Wide disk (2 spares)	2	232916-B22	$605	2	$1,210	
Hardware Support 3 yr, 24x7, 4hr empty enclosure	2	171242-002	$157	21		$3,297
	Server Subtotal				**$368,081**	**$13,808**
Microsoft Windows .NET Enterprise Server 2003	4	N/A	$5,399	1	$5,399	
Microsoft SQL Server 2000 Enterprise Edition 64 bit	4	810-00560	$16,541	4	$66,164	
HP Support for Windows Advanced Server 3 yr 24x7	1	H4405Y#6BR	$7,302	1		$7,302
MS Software Support (3 yrs)	4	N/A	$5,850	1		$5,850
	Server Software Subtotal				**$71,563**	**$13,152**
HP ProLiant DL360R01 P1.4GHz 512KB 128MB	2	233271-001	$2,229	5	$11,145	
1.40GHz PIII Processor Option Kit (DL360 G2)	2	233273-B21	$734	5	$3,670	
1G PC133-MHz option Kit	2	201694-B21	$425	5	$2,125	
HP Mouse	2	261602-001	$5	5	$25	
HP Enhanced Keyboard	2	265977-001	$12	5	$60	
S5500 15 carbon / silver monitor	2	261602-001	$139	5	$695	
18GB, 15krpm Ultra3 Wide disk	2	188122-B22	$390	5	$1,950	
Qlogic QLA-2350 Fibre-Channel VI Adapter	3	QLA2350-BK	$2,095	5	$10,475	
5M LC to LC Cable Kit	2	221692-B22	$82	5	$410	
2GB Small Form Pluggable Adapter Kit	2	221470-B21	$369	5	$1,845	
FM-EL724-36 3YR 24X7 4HR 300 SERIES SVR	2	162657-002	$1,450	5		$7,250
	Client Subtotal				**$32,400**	**$7,250**
Microsoft Windows 2000 Server	4	C11-00821	$738	5	$3,690	
Microsoft Visual C++ .NET Standard	4	254-00170	$109	1	$109	
	Client Software Subtotal				**$3,799**	
Qlogic SANBox2-8 8-Port Switch	3	SANBOX 2-8	$7,995	1	$7,995	
	Connectivity Subtotal				**$7,995**	
HP's Large Configuration Discount *	*** Discounts:**				**-$71,420**	**-$5,606**
	Total:				**$412,418**	**$28,604**
Price Key: 1-HP at 22% discount, 2-HP at 16% discount, 3-Qlogic, 4-Microsoft	**3 year cost of ownership:**					**$441,022**
	tpmC:					87741
* All discounts are based on US list prices and for similar quantities and configurations	**$/tpmC:**					**$5.03**

discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specifications. If you find that the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.

Results independently audited by Francois Raab of InfoSizing, Inc. (www.sizing.com)

Numerical Quantities Summary for hp server rx5670

MQTH, Computed Maximum Qualified Throughput **87741.45 tpmC**

Response Times (in seconds)

	90th %-ile	Maximum	Average
New-Order	0.67s	11.54s	0.39s
Payment	0.62s	12.09s	0.34s
Order-Status	0.64s	9.20s	0.35s
Delivery (interactive portion)	0.11s	0.13s	0.10s
Delivery (deferred portion)	0.16s	1.64s	0.12s
Stock-Level	0.99s	9.72s	0.62s
Menu	0.11s	0.59s	0.10s

Response time delay added for emulated components 0.1 seconds

Transaction Mix, in percent of total transactions

New-Order	44.95%
Payment	43.02%
Delivery	4.01%
Stock-Level	4.01%
Order-Status	4.01%

Keying/Think Times

	Keying Time			Think Time		
	Min	Avg	Max	Min	Avg	Max
New-Order	18s	18.02s	18.03s	0s	12.05s	120.51s
Payment	3s	3.02s	3.03s	0s	12.05s	120.51s
Order-Status	2s	2.02s	2.02s	0s	10.04s	100.5s
Delivery (interactive)	2s	2.02s	2.02s	0s	5.06s	50.5s
Stock-Level	2s	2.02s	2.02s	0s	5.04s	50.5s

Test Duration

Ramp up time	28 minutes
Measurement interval	120 minutes
Transactions during measurement interval	24363926
Ramp down time	9 minutes

Checkpointing

Number of checkpoints in measurement interval	4
Checkpoint Interval	30 minutes

Table of Contents

Preface

Document Structure

This is the full disclosure report for a benchmark test of the hp server rx5670 using Microsoft SQL Server 2000 Enterprise Edition 64-bit. It meets the requirements of the TPC Benchmark ® C Standard Specification, Revision 5.0 dated February 26, 2001. TPC Benchmark® C was developed by the **T**ransaction Processing **P**erformance **C**ouncil (TPC). It is the intent of this group to develop a suite of benchmarks to measure the performance of computer systems executing a wide range of applications. Hewlett-Packard Company and Microsoft, Inc. are active participants in the TPC.

TPC Benchmark® C Overview

TPC Benchmark ® C is an **O**n **L**ine **T**ransaction **P**rocessing (OLTP) workload. It is a mixture of read-only and update intensive transactions that mimic the activities found in complex OLTP application environments. It does so by exercising a breadth of system components associated with such environments, which are characterized by:

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

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

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

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

Hewlett-Packard Company does not warrant or represent that a user can or will achieve performance similar to the benchmark results contained in this report. No warranty of system performance or price/performance is expressed or implied by this report.

System Overview

The hardware configuration used in this TPC-C test was based on the hp server rx5670. The full configuration was built by adding additional memory, additional disk adapters and drives. The operating system used on the server was Microsoft Windows .NET Enterprise Server 2003 and the database was Microsoft SQL Server 2000 Enterprise Edition 64-bit.

The processor architecture of the hp server rx5670 was designed for the Intel Itanium 2 processor. The hp server rx5670 used in this test was powered by four 1 GHz Intel Itanium 2 processors, each with 3MB of 3rd level cache.

This configuration used 48 GB of HP DDRAM. This was achieved by using 48 1 GB DIMMs.

The operating system, all executables and libraries, the master database, and swap space were contained in one 36GB hard disk, attached to an embedded PCI SCSI controller.

This measured configuration used six hp SmartArray 5304 Disk Array Controllers (DACs), which were plugged into PCI-X slots. The database log drives storage consisted of 14 36GB 15krpm Ultra-160 SCSI hard drives attached to one DAC. The disks were configured as RAID 0+1. The Array Accelerator (Cache) was disabled on the log DAC.
The TPC-C database storage consisted of 280 HP 18GB 15krpm hard drives. The 18GB drives on each DAC were equally distributed across 4 SCSI channels (14 per channel). The Disks were configured as 1 RAID 0 volume over 56 disk on each DAC. The Array Accelerator (Cache) was enabled on the database DACs and configured for 100% write.

Each of the five clients is a HP Proliant DL360 with two Intel Pentium III at 1.4GHz, 1024 MB RAM and one 18.1GB SCSI hard disk, running Microsoft Windows 2000 Server.

The server and web-clients were networked together using QLogic VIA/SAN. The network bandwith between the web-clients and the database server was 2GBps. 10 remote terminal emulators (RTEs) emulated 70,000 users executing the standard TPC-C workload. Each web-client had two embedded Gigabit LAN adapters, one of which was was used to connect to the RTEs running in 100MBit mode.

General Items

Test Sponsor

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

Hewlett-Packard Company was the test sponsor of this TPC Benchmark C.

Application Code and Definition Statements

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

The Section 3.0 entitled Clause 3 Related Items contains a brief discussion of the database design and loading. The database definition statements, distribution across disk drives, loading scripts, and tables are provided in Appendix B.

The program that implements the TPC Benchmark C translation and collects appropriate transaction statistics is referred to as the Remote Terminal Emulator (RTE) or Driver program. We have used the Microsoft BenchCraft RTE program that emulated a set of users entering TPC-C transactions through web browsers, and communicating with web-client machines running the Microsoft Internet Information Server (IIS) web server. The web-client machines used the COM+ transaction monitor (TM) to communicate with the database server.

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

Parameter Settings

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

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

Appendix C contains all the database and operating system parameters used in this benchmark in addition to all the hardware configuration details.

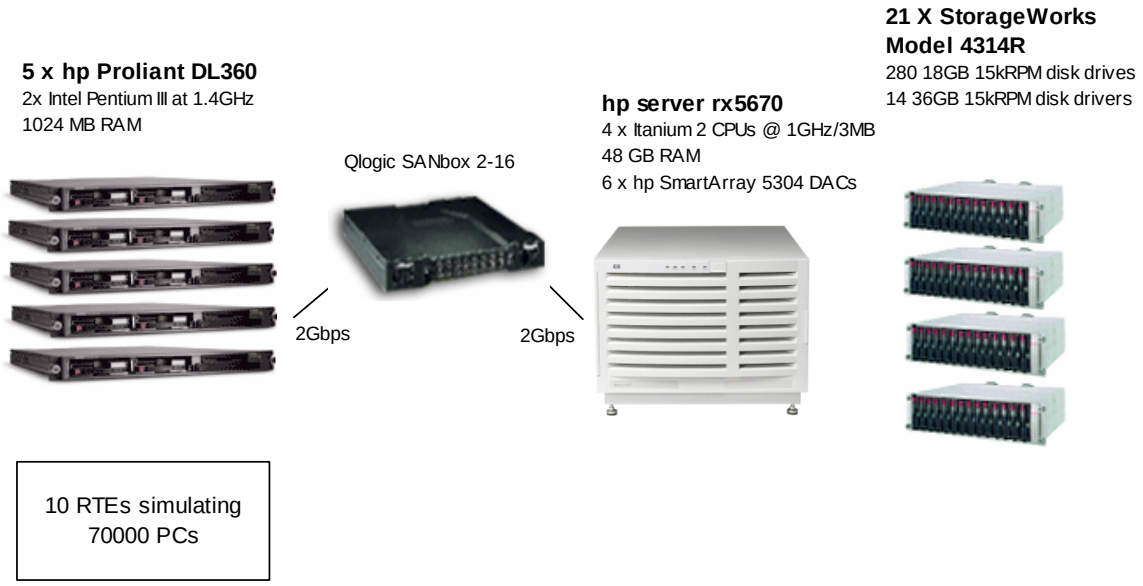
Appendix D contains the 60 day space calculations.

Configuration Diagrams

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

The measured and priced client/server configuration is shown in Figures 1.

Figure 1. Measured and Priced Configuration



Chapter 1 Logical Database Design

1.1 Table Definitions

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

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

1.2 Physical Organization of the Database

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

The measured database configuration used a total of 295 disks, which included 280 18GB Hot Swap disk drives for data, 14 18GB drives for log, and one 18GB drive for the operating system.
Each of the 5 database DACs was Configured as 1 RAID 0 volume over 56 18GB drives. Each volume held 3 partitions,
One for the CS filegroup where the Customer and Stock tables were stored and one partition for MISC filegroup where all other tables were stored. The third partition was used to store backups of the database.

1.3 Insert and Delete Operations

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

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

1.4 Partitioning

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

Partitioning was not used on any table.

1.5 Replication, Duplication or Additions

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

No replications, duplications or additional attributes were used.

Chapter 2 Transaction and Terminal Profiles

2.1 Random Number Generation

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

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

2.2 Input/Output Screen Layout

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

The screen layouts are based on those in Clauses 2.4.3, 2.5.3, 2.6.3, 2.7.3, and 2.8.3 of the TPC-C® Standard Specification.

2.3 Priced Terminal Feature Verification

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

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

2.4 Transaction Statistics

The transaction profiles must be disclosed as per Clauses 8.1.3.5 through 8.1.3.10.

Table 1 shows the transaction statistics.

Table 1. Transaction Statistics

Type	Item	Value
New Order	Home warehouse items	99.00%
	Remote warehouse items	1.00%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse	85.00%
	Remote warehouse	15.00%
	Non primary key access	59.99%
Order Status	Non primary key access	60.06%
Delivery	Skipped transactions	0
Transaction Mix	New Order	44.95%
	Payment	43.02%
	Delivery	4.01%
	Stock Level	4.01%
	Order Status	4.01%

2.5 Presentation Manager or Intelligent Terminal

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

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

Application code running on the web-client implemented the TPC-C® user interface. Screen manipulation commands in the form of HTML were downloaded to the web browser, which handled input and output presentation graphics. A listing of this code is included in Appendix A. Microsoft Internet Information Service assisted in the processing and presentation of this data.

2.6 Queuing Mechanism

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

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

The source code is listed in Appendix A.

Chapter 3 Transaction and System Properties

3.1 Transaction System Properties (ACID Tests)

Results of the ACID test must describe how the requirements were met. This includes disclosing which case was followed for the execution of Isolation Test 7.

The TPC Benchmark C standard specification defines a set of transaction processing system properties that a System Under Test (SUT) must support during the execution of the benchmark. Those properties are Atomicity, Consistency, Isolation and Durability (ACID). The following subsections will define each of these properties and describe the series of tests that were performed by HP to demonstrate that the properties were met.

All of the specified ACID tests were successfully performed on the hp server rx5670. A fully scaled database was used for all the durability tests.

3.2 Atomicity Tests

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

3.2.1 COMMIT Transaction

The following steps were followed to demonstrate the COMMIT property of Atomicity:

A row was randomly selected from the Warehouse, District and Customer tables, and the present balances noted. The standard payment transaction was started against the above identifiers using a known amount. The transaction was committed and the rows were verified to contain the correct updated balances.

3.2.2 ROLLBACK Transaction

The following steps were followed to demonstrate the COMMIT property of Atomicity:

A row was randomly selected from the Warehouse, District and Customer tables, and the present balances noted. The standard payment transaction was started against the above identifiers using a known amount. The transaction was rolled back and the rows were verified to contain the original balances.

3.3 Consistency Tests

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

Consistency conditions 1 through 4 were tested using a shell script to issue queries to the database. The results of the queries verified that the database was consistent for all four tests. A performance run was executed at rated speed. The shell script was executed again. The result of the same queries verified that the database remained consistent after the run.

3.4 Isolation Tests

Operations of concurrent transactions must yield results which are indistinguishable from the results which would be obtained by forcing each transaction to be serially executed to completion in some order.

This property is commonly called serializability. Sufficient conditions must be enabled at either the system or application level to ensure serializability of transactions under any mix of arbitrary transactions.

We ran a total of nine isolation tests. Seven of these tests are detailed in the TPC-C specification (clause 3.4.2.1to 3.4.2.7). The additional two are to fully comply with the isolation requirements that are not directly specified in the TPC-C specification. These two tests are known as Phantom Protection One and Two. They demonstrate that the applications are protected from phantom inserts.

3.5 Durability Tests

The tested system must guarantee the ability to preserve the effects of committed transactions and insure database consistency after recovery from any one of the failures listed in clause 3.5.3.1, 3.5.3.2, and 3.5.3.3.

Three types of failures were tested to ensure the durability of the database: Loss of Data, Loss of Log, and Loss of System/Memory.
All test were performed on the full scale database..

3.5.1 Loss of Data

The standard driving mechanism was used to generate the transaction load of 7000 users for the test. To demonstrate recovery from a permanent failure of durable media containing TPC-C tables, the following steps were executed:

1. The database was backed up using SQLServer backup facilities.
2. A sum of D_NEXT_O_ID was taken.
3. 7000 users were logged in to the database and ran transactions.
4. After 5 minutes, one data disk drive was removed causing SQLServer errors.
5. The RTE was allowed to continue running. Completed transactions enroute from the clients were recorded. Error messages started appearing on the RTE screen.
6. All users were paused and stopped from the RTE.
7. SQLServer was stopped and restarted and a dump of the transaction log was taken.
8. SQLServer was stopped, Windows was shutdown, and the machine was powered off.
9. The failed disks were replaced and the controller configuration utility was run to make the two disks ‘online’.
10. The machine was powered up, Windows and SQLServer were started.
11. The TPC-C database was dropped and restored from the backup.
12. The transaction log was restored and transactions rolled forward.
13. A new count of D_NEXT_O_ID was taken.
14. This number was compared with the number of new orders reported by the RTE.
15. Samples were taken of the RTE log and verified against the database.

3.5.2 Loss of System / Memory and loss of Log

This was demonstrated on the full database with 7000 warehouses in a single test. The standard driving mechanism was used to generate the transaction load of 70,000 users for this test. To demonstrate recovery the following steps were followed:

1. The full database was used.
2. A sum of D_NEXT_O_ID was taken.
3. 70,000 users were logged in to the database and ran transactions.
4. After 5 minutes, one of the (mirrored) log disk was removed from the system, processing transactions continued
5. After another 5 minutes, power to the system was cut off.
6. The RTE continued running and completed transactions enroute from the clients were recorded. Error messages began appearing on the RTE screen.
7. The RTE was stopped.
8. The server machine was powered on again and rebooted.
9. Microsoft SQL Server was restarted and performed an automatic recovery.
10. A new count of D_NEXT_O_ID was taken.
11. This number was compared with the number of new orders reported by the RTE
12. Samples were taken of the RTE log and verified against the database.

Chapter 4 Scaling and Database Population

4.1 Database Layout

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

The measured (tested) and priced systems used 6 HP SmartArray 5304 4-channel PCI Disk Array Controllers (DACs). These cards plugged into PCI-X slot 4, 6, 8, 10, 11 and 12 on the I/O backplane. The system drive was attached to the embedded Ultra3 LVD SCSI channel.

The measured database configuration used a total of 295 disks, which included 280 18GB Hot Swap disk drives for data, 14 18GB drives for log, and one 18GB drive for the operating system.
Each of the 5 database DACs was Configured as 1 RAID 0 volume over 56 18GB drives. Each volume held 3 partitions,
One for the CS filegroup where the Customer and Stock tables were stored and one partition for MISC filegroup where all other tables were stored. The third partition was used to store backups of the database.

Table 2 shows the complete data distribution.

Table 2: Data Distribution

HP SmartArray 5304					WINDOWS.NET DISK ADMINISTRATION
Controller #1					Disk 0: 237 GB
SCSI ID	Channels				Partition (RAID 0+1)
	0	1	2	3	1
0	36GB				L: LOG 237 GB
1	36GB				
2	36GB				
3	36GB				
4	36GB				
5	36GB				
6	36GB				
8		36GB			
9		36GB			
10		36GB			
11		36GB			
12		36GB			
13		36GB			
14		36GB			

HP SmartArray 5304					WINDOWS.NET DISK ADMINISTRATION		
Controller #2					Disk 1: 950 GB		
SCSI ID	Channels				Partitions (RAID 0)		
	0	1	2	3	1	2	3
0	18GB	18GB	18GB	18GB	C:\MNT\CS1 raw 80 GB	C:\MNT\MISC1 raw 41 GB	C:\MNT\FS1 NTFS 829 GB
1	18GB	18GB	18GB	18GB			
2	18GB	18GB	18GB	18GB			
3	18GB	18GB	18GB	18GB			
4	18GB	18GB	18GB	18GB			
5	18GB	18GB	18GB	18GB			
6	18GB	18GB	18GB	18GB			
8	18GB	18GB	18GB	18GB			
9	18GB	18GB	18GB	18GB			
10	18GB	18GB	18GB	18GB			
11	18GB	18GB	18GB	18GB			
12	18GB	18GB	18GB	18GB			
13	18GB	18GB	18GB	18GB			
14	18GB	18GB	18GB	18GB			

HP SmartArray 5304					WINDOWS.NET DISK ADMINISTRATION		
Controller #3					Disk 2: 950 GB		
SCSI ID	Channels				Partitions (RAID 0)		
	0	1	2	3	1	2	3
0	18GB	18GB	18GB	18GB	C:\MNT\CS2 raw 80 GB	C:\MNT\MISC2 raw 41 GB	C:\MNT\FS2 NTFS 829 GB
1	18GB	18GB	18GB	18GB			
2	18GB	18GB	18GB	18GB			
3	18GB	18GB	18GB	18GB			
4	18GB	18GB	18GB	18GB			
5	18GB	18GB	18GB	18GB			
6	18GB	18GB	18GB	18GB			
8	18GB	18GB	18GB	18GB			
9	18GB	18GB	18GB	18GB			
10	18GB	18GB	18GB	18GB			
11	18GB	18GB	18GB	18GB			
12	18GB	18GB	18GB	18GB			
13	18GB	18GB	18GB	18GB			
14	18GB	18GB	18GB	18GB			

HP SmartArray 5304					WINDOWS.NET DISK ADMINISTRATION		
Controller #4					Disk 3: 950 GB		
SCSI ID	Channels				Partitions (RAID 0)		
	0	1	2	3	1	2	3
0	18GB	18GB	18GB	18GB	C:\MNT\CS3 raw 80 GB	C:\MNT\MISC3 raw 41 GB	C:\MNT\FS3 NTFS 829 GB
1	18GB	18GB	18GB	18GB			
2	18GB	18GB	18GB	18GB			
3	18GB	18GB	18GB	18GB			
4	18GB	18GB	18GB	18GB			
5	18GB	18GB	18GB	18GB			
6	18GB	18GB	18GB	18GB			
8	18GB	18GB	18GB	18GB			
9	18GB	18GB	18GB	18GB			
10	18GB	18GB	18GB	18GB			
11	18GB	18GB	18GB	18GB			
12	18GB	18GB	18GB	18GB			
13	18GB	18GB	18GB	18GB			
14	18GB	18GB	18GB	18GB			

HP SmartArray 5304					WINDOWS.NET DISK ADMINISTRATION		
Controller #5					Disk 4: 950 GB		
SCSI ID	Channels				Partitions (RAID 0)		
	0	1	2	3	1	2	3
0	18GB	18GB	18GB	18GB	C:\MNT\CS4 raw 80 GB	C:\MNT\MISC4 raw 41 GB	C:\MNT\FS4 NTFS 829 GB
1	18GB	18GB	18GB	18GB			
2	18GB	18GB	18GB	18GB			
3	18GB	18GB	18GB	18GB			
4	18GB	18GB	18GB	18GB			
5	18GB	18GB	18GB	18GB			
6	18GB	18GB	18GB	18GB			
8	18GB	18GB	18GB	18GB			
9	18GB	18GB	18GB	18GB			
10	18GB	18GB	18GB	18GB			
11	18GB	18GB	18GB	18GB			
12	18GB	18GB	18GB	18GB			
13	18GB	18GB	18GB	18GB			
14	18GB	18GB	18GB	18GB			

HP SmartArray 5304					WINDOWS.NET DISK ADMINISTRATION		
Controller #6					Disk 5: 950 GB		
SCSI ID	Channels				Partitions (RAID 0)		
	0	1	2	3	1	2	3
0	18GB	18GB	18GB	18GB	C:\MNT\CS5 raw 80 GB	C:\MNT\MISC5 raw 41 GB	C:\MNT\FS5 NTFS 829 GB
1	18GB	18GB	18GB	18GB			
2	18GB	18GB	18GB	18GB			
3	18GB	18GB	18GB	18GB			
4	18GB	18GB	18GB	18GB			
5	18GB	18GB	18GB	18GB			
6	18GB	18GB	18GB	18GB			
8	18GB	18GB	18GB	18GB			
9	18GB	18GB	18GB	18GB			
10	18GB	18GB	18GB	18GB			
11	18GB	18GB	18GB	18GB			
12	18GB	18GB	18GB	18GB			
13	18GB	18GB	18GB	18GB			
14	18GB	18GB	18GB	18GB			

4.2 Initial Cardinality of Tables

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

Table 2 shows the cardinality of the various tables.

Table 3: Table Cardinality

Table	Occurrences
Warehouse	7,000
District	70,000
Customer	210,000,000
History	210,000,000
Orders	210,000,000
New Orders	63,000,000
Order Line	2,099,998,534
Stock	100,000
Item	700,000,000

No rows were deleted for the benchmark runs.

4.3 60 Day Space

Details of the 60 day space computations along with proof that the database is configured to sustain 8 hours of growth for the dynamic tables must be disclosed.

4.3.1 Transaction Log Space Requirements

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

1. The free space on the logfile was queried using **dbcc sqlperf(logspace)**.

2. Transactions were run against the database with a full load of users.

3. The free space was again queried using **dbcc sqlperf(logspace)**.

4. The space used was calculated as the difference between the first and second query.

5. The number of NEW-ORDERS was verified from an RTE report covering the entire run.

6. The space used was divided by the number of NEW-ORDERS giving a space used per NEW-ORDER transaction.

7. The space used per transaction was multiplied by the measured tpmC rate times 480 minutes.

The result of the above steps yielded a requirement of 191.58 GB to sustain the log for 8 hours. Space available for the transaction log was 237 GB, indicating that enough storage was configured to hold 8 hours of growth.

The same methodology was used to calculate the growth requirements for the other dynamic tables Order, Order-Line and History. The details of the 60ay growth calculation are shown in Appendix D.

4.4 Type of Database Used

A statement must be provided that describes:

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

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

Microsoft SQL Server 2000 Enterprise Edition 64-bit is a relational DBMS.

The interface was SQL Server stored procedures accessed with library calls embedded in C code.

4.5 Database Mapping

The mapping of database partitions and replications must be described.

The database was divided into 2 file groups MSSQL70_misc_fg and MSSQL70_cs_fg. MSSQL70_misc_fg consists of 5 partitions at 41,500 MB each and MSSQL70_misc_fg consist of 5 partitions at 81,200 MB each as shown in the createdb.sql. The log was configured with 90,000 MB.

Chapter 5 Performance Metrics and Response Time

5.1 Throughput

Measured tpmC® must be reported.

Measured TpmC®: 87,741.45
Price per TpmC®: $5.03

5.2 Response Times

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

Table 3 shows the response times for all transaction types.

Table 4: Transaction Response Times

Response Times	Average	90th %-ile	Maximum
New-Order	0.39s	0.67s	11.54s
Payment	0.34s	0.62s	12.09s
Order-Status	0.35s	0.64s	9.20s
Delivery (interactive portion)	0.10s	0.11s	0.13s
Delivery (deferred portion)	0.12s	0.16s	1.64s
Stock-Level	0.62s	0.99s	9.72s
Menu	0.10s	0.11s	0.59s

5.3 Keying and Think Times

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

Tables 4 and 5 show the key times and think times for all transaction types.

Table 5: Transaction Key Times

Keying Times	Minimum	Average	Maximum
New Order	18s	18.02s	18.03s
Payment	3s	3.02s	3.03s
Order Status	2s	2.02s	2.02s
Interactive Delivery	2s	2.02s	2.02s
Stock Level	2s	2.02s	2.02s

Table 6: Transaction Think Times

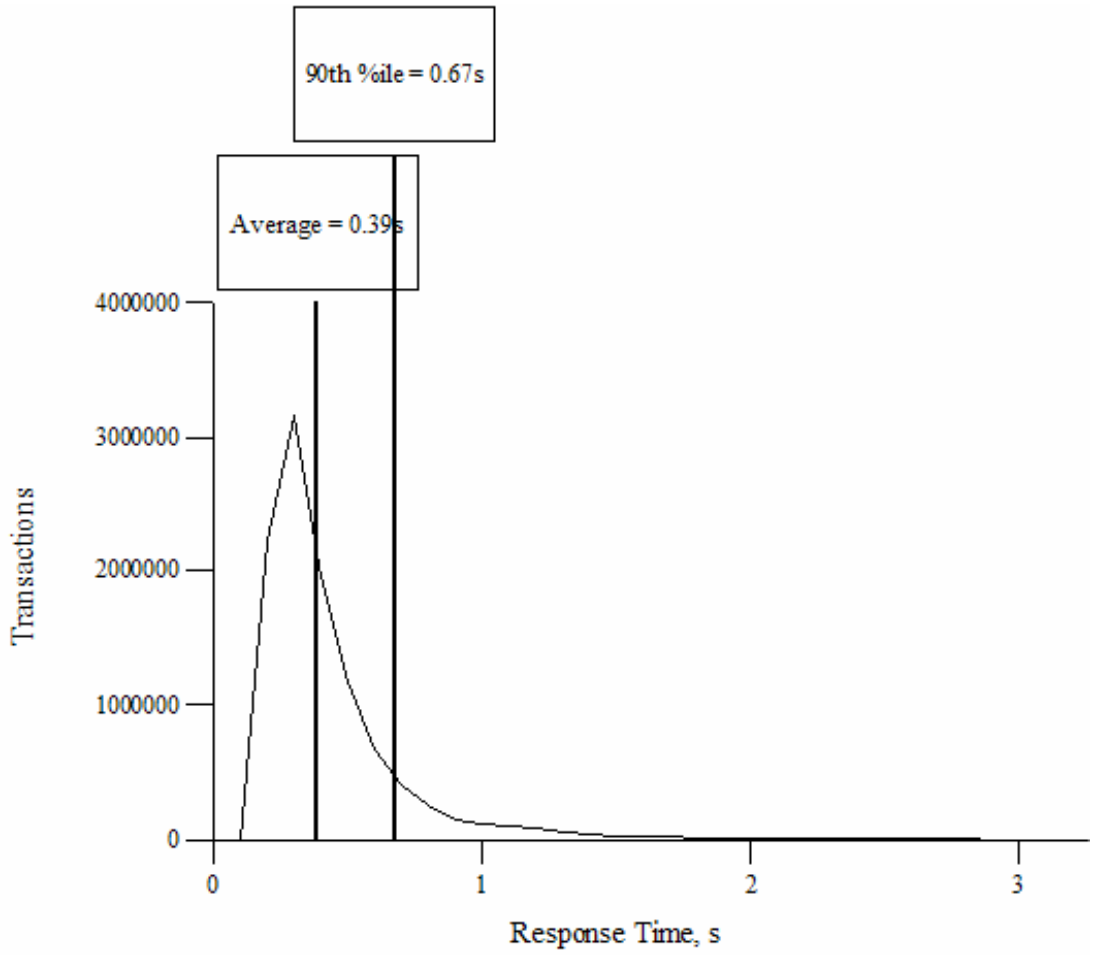
Think Times	Minimum	Average	Maximum
New Order	0s	12.05s	120.51s
Payment	0s	12.05s	120.51s
Order Status	0s	10.04s	100.5s
Interactive Delivery	0s	5.06s	50.5s
Stock Level	0s	5.04s	50.5s

5.4 Response Time Frequency

Response Time frequency distribution curves (see Clause 5.6.1) must be reported for each transaction type. The performance curve for response times versus throughput (see Clause 5.6.2) must be reported for the New-Order transaction. Think Time frequency distribution curves (see Clause 5.6.3) must be reported for each transaction type. Keying Time frequency distribution curves (see Clause5.6.4) must be reported for each transaction type. A graph of throughput versus elapsed time (see Clause 5.6.5) must be reported for the New-Order transaction.

5.4.1 New Order Response Time

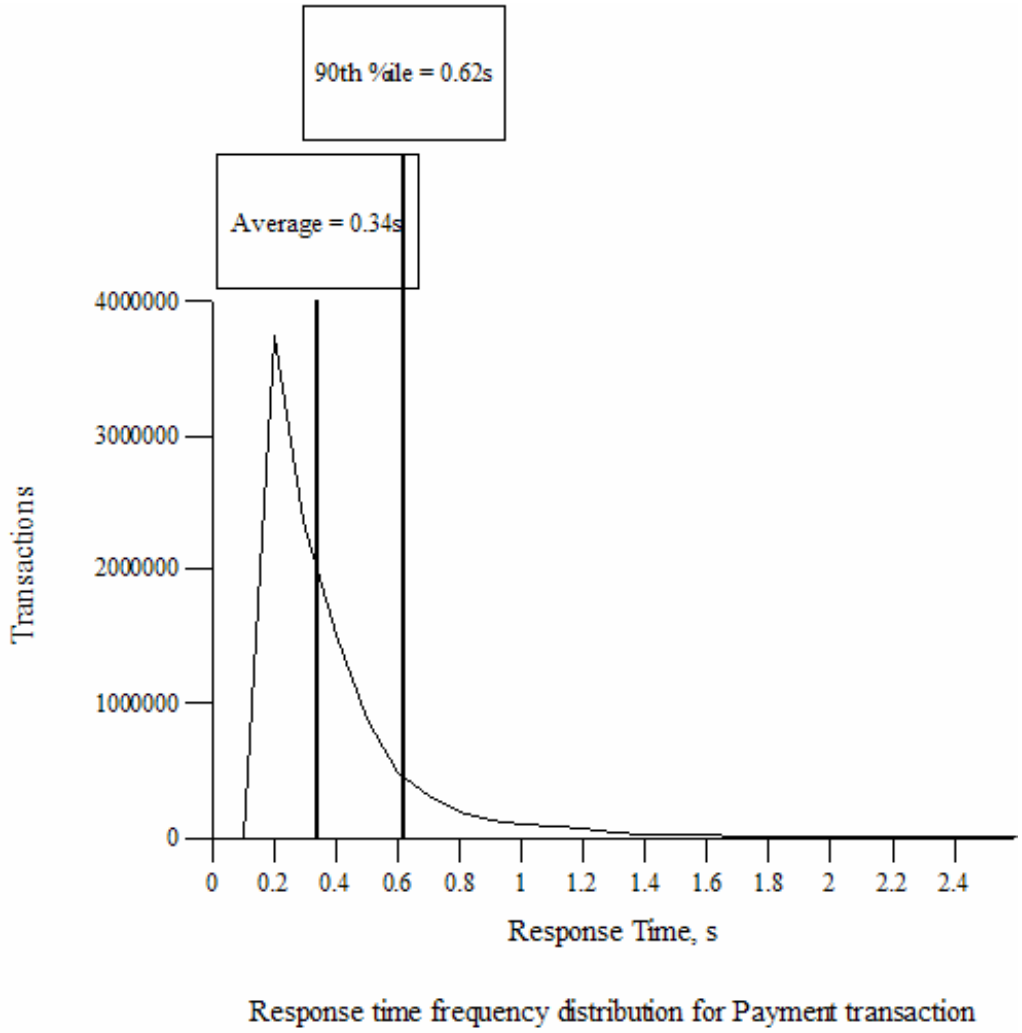
Figure 3: New Order Response Time Distribution



Response time frequency distribution for New Order transaction

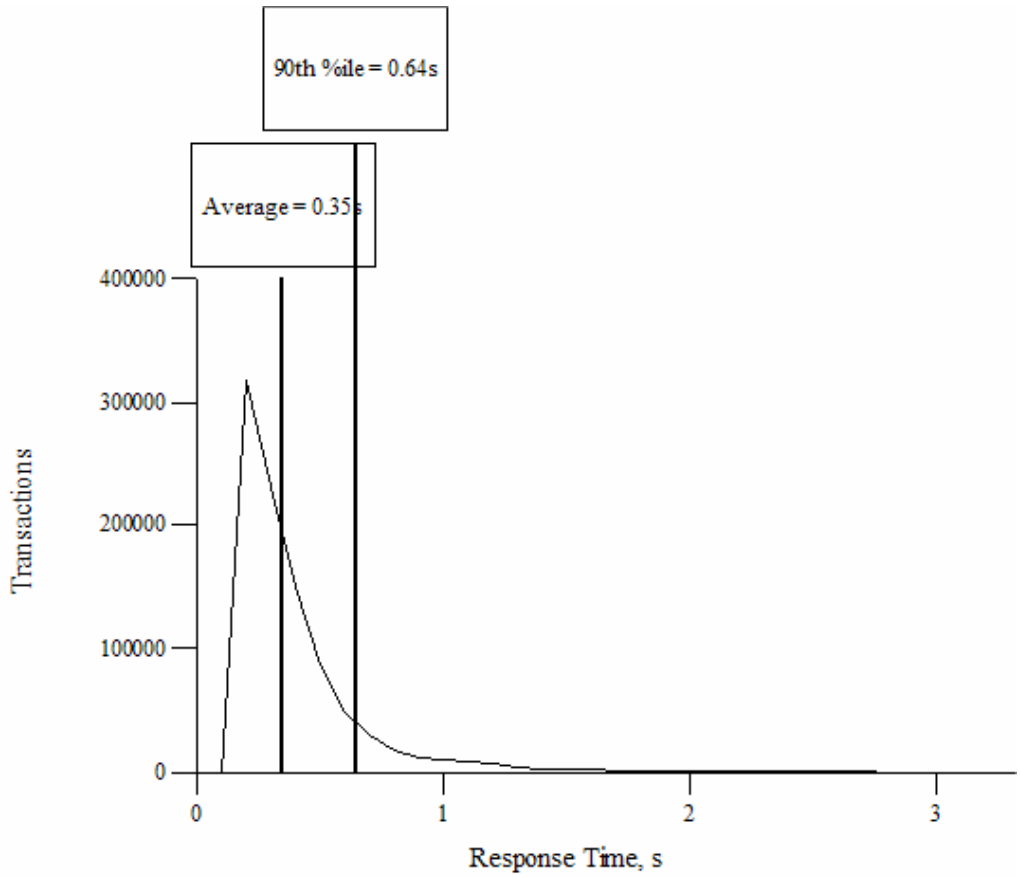
5.4.2 Payment Response Time Distribution

Figure 4: Payment Response Time Distribution



5.4.3 Order Status Response Time

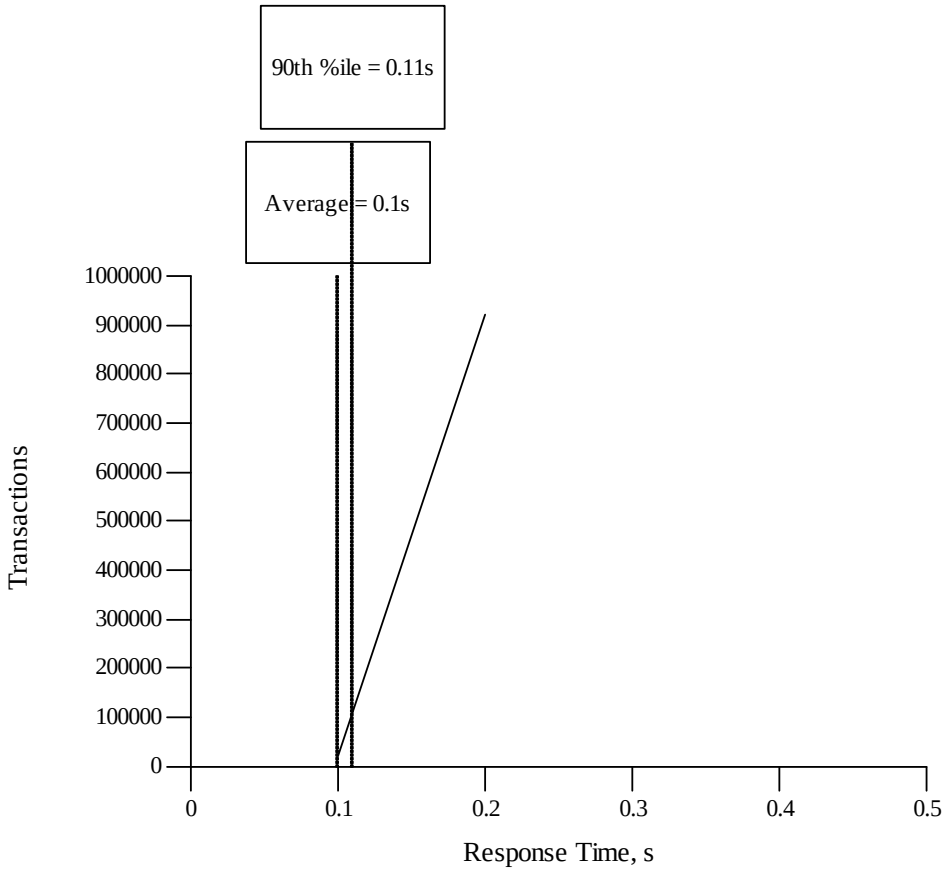
Figure 5: Order Status Response Time Distribution



Response time frequency distribution for Order Status transaction

5.4.4 Delivery Response Time Distribution

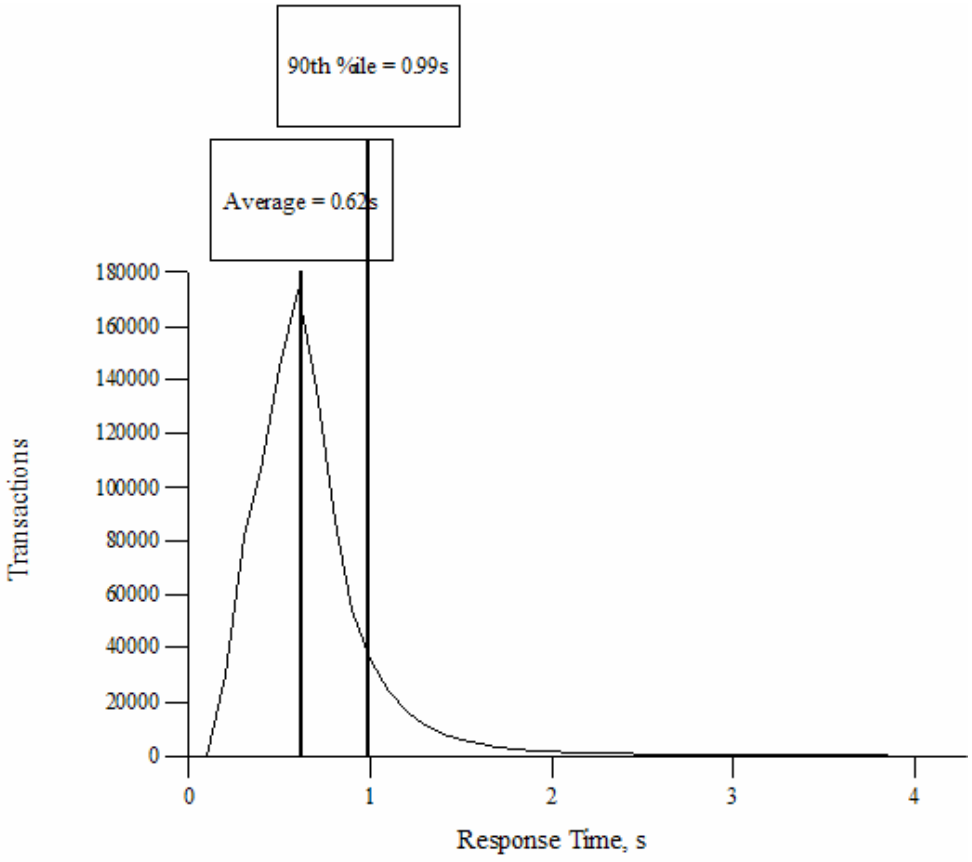
Figure 6: Delivery Response Time Distribution



Response time frequency distribution for Delivery transaction

5.4.5 Stock Level Response Time

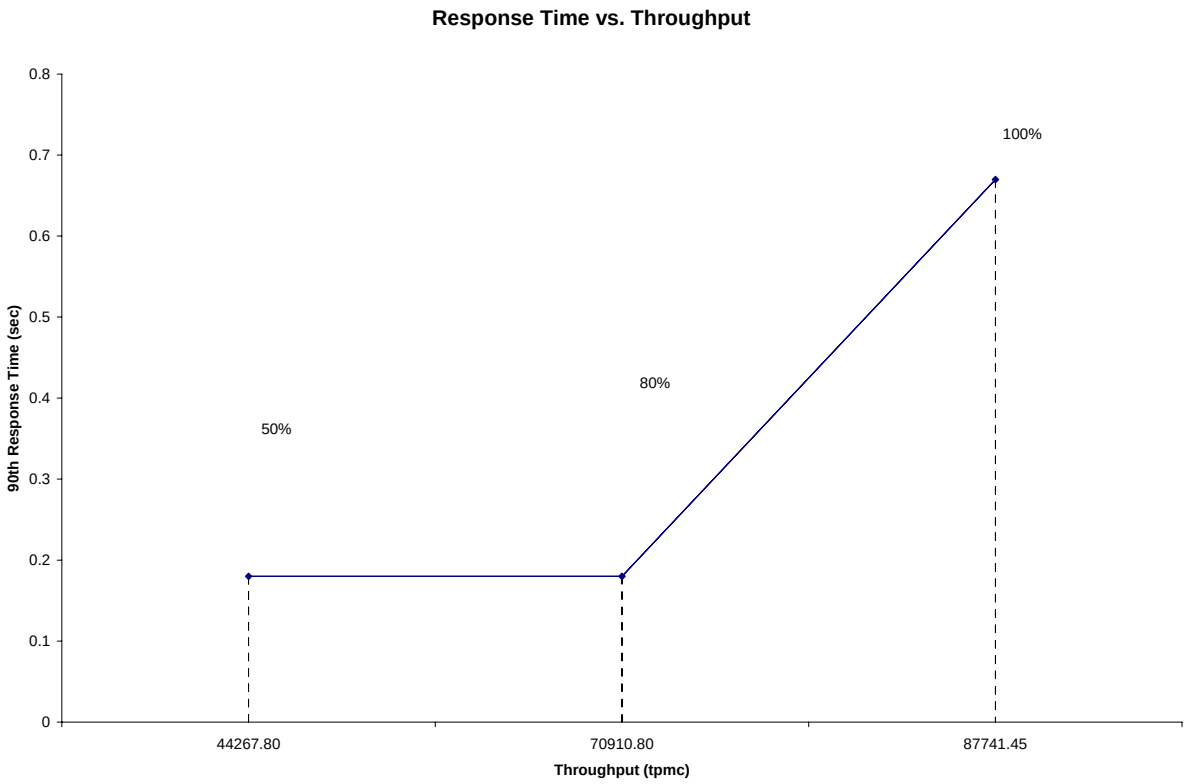
Figure 7: Stock Level Response Time Distribution



Response time frequency distribution for Stock Level transaction

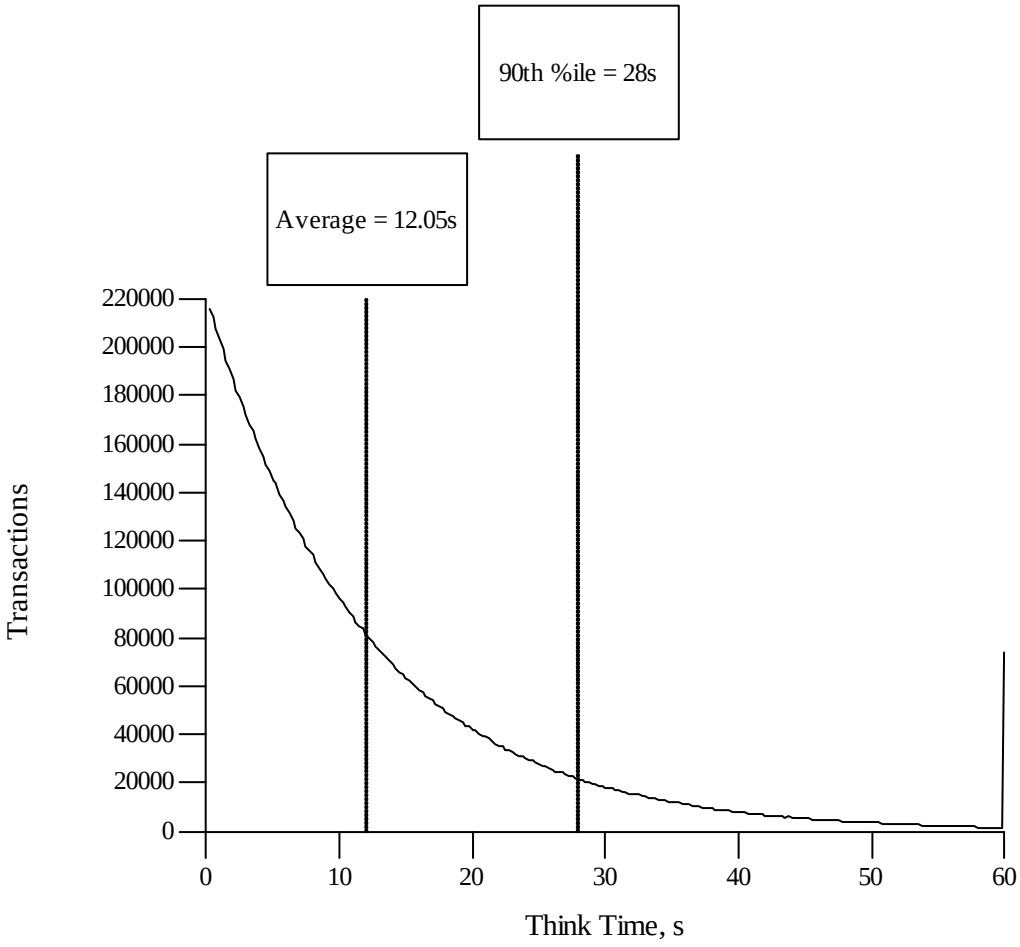
5.4.6 Response Time Versus Throughput

Figure 8: New Order Response Time Distribution



5.4.7 New Order Think Time Distribution

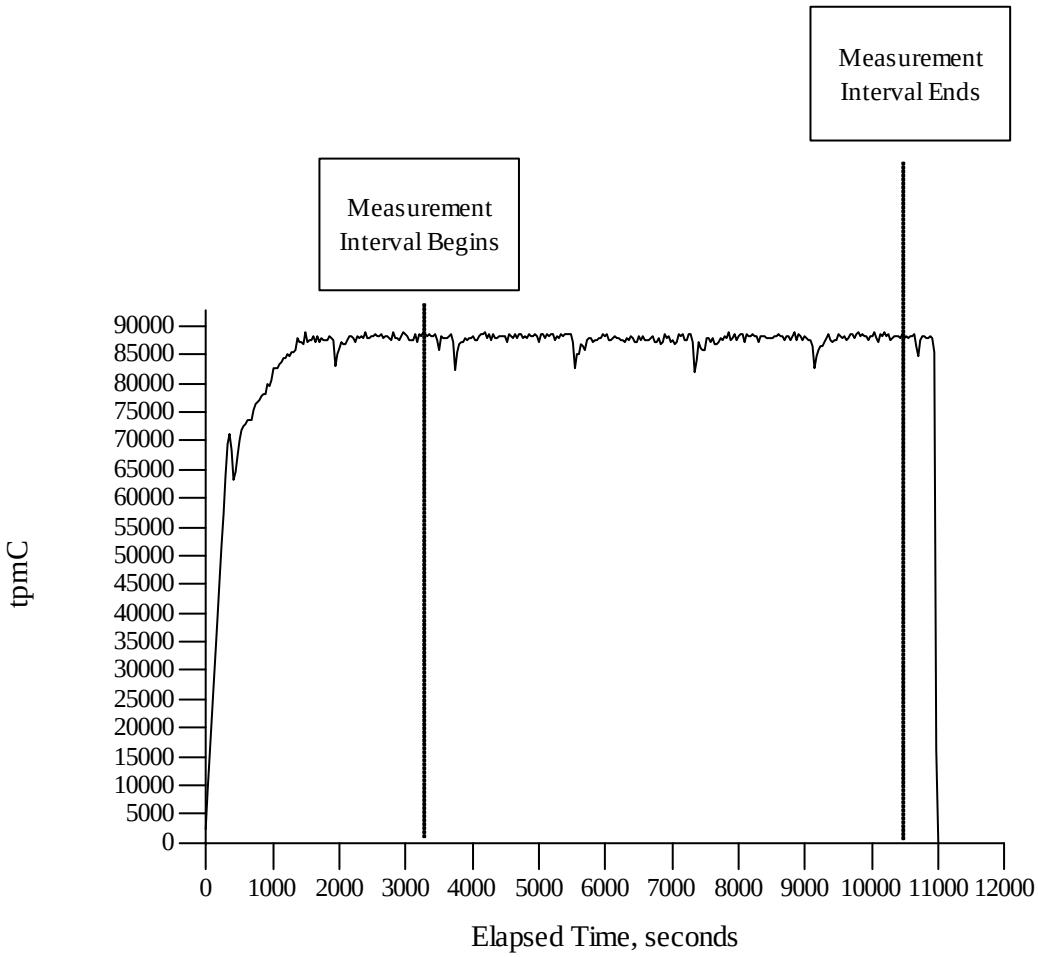
Figure 9: New Order Think Time Distribution



Think time frequency distribution for New Order transaction

5.4.8 Throughput Versus Time Distribution

Figure 10: New Order Throughput versus Time



Throughput of the New-Order transaction versus elapsed time

5.5 Steady State Determination

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

The transaction throughput rate (tpmC®) and response time were relatively constant after the initial 'ramp up' period. The throughput and response time behaviors were determined by examining data reported for each interval over the duration of the benchmark. The corresponding graph is in Figure 10.

5.6 Work Performed During Steady State

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

The RTEs generated the required input data to choose a transaction from the menu. This data was timestamped. The menu response time for the requested transaction was verified and timestamped in the RTE log files. The RTE generated the required input data for the chosen transaction. It waited to complete the minimum required key time before transmitting the HTTP request to the client. The transmission was timestamped. The return of the screen with the required response data was timestamped. The difference between these two timestamps was the response time for that transaction and was logged in the RTE log. The RTE then waited the required think time interval before repeating the process and starting another transaction.

5.6.1 Checkpoint

The checkpoint mechanism is an automatic means for guaranteeing that completed transactions are regularly written from SQL Server's disk cache to the database device. A checkpoint writes all "dirty pages"-cached pages that have been modified since the last checkpoint-to the database device.

5.6.2 Checkpoint Conditions

There are two types of checkpoints:

1. Checkpoints that are executed automatically by SQL Server.

2. Checkpoints that are forced by database owners with the CHECKPOINT statement.

Forcing dirty pages onto the database device means that all completed transactions are written out. By calling all completed transactions to be written out, the check point shortens the time it takes to recover, since the database pages are current and there are no transactions that need to be rolled forward.

5.6.3 Checkpoint Implementation

To perform checkpoints at specific intervals, the SQL Server *recovery interval* was set to 56 and a NT command script was issued to schedule multiple checkpoints at specific intervals. The manual checkpoints were spread across 25% of the recovery interval, which was 14 minutes. During the ramp-up and after all users were active, a background process slept and performed the checkpoint every 30 minutes. The measurement interval was equal to 4 times the waiting time between the checkpoints. By setting the TRACE FLAG #3502, SQL Server logged the checkpoint beginning and ending time in the ERRORLOG file.

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

5.7 Measurement Period Duration

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

The measurement interval was 120 minutes.

5.8 Regulation of Transaction Mix

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

The weighted average method of *Clause 5.2.4.1* was used. The weights were not adjusted during the run.

5.9 Transaction Mix

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

Table 7: Transaction Mix

Type	Percentage
New Order	44.95%
Payment	43.02%
Delivery	4.01%
Stock Level	4.01%
Order Status	4.01%

5.10 Transaction Statistics

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

Table 1contains the required items.

5.11 Checkpoint Count and Location

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

The measurement interval is 120 minutes. There are 4 checkpoints within the measurement interval and one checkpoint before the measurement interval.

Table 8: Measurement Interval and Checkpoints

Event	From (s)	To (s)	Duration (s)
Measured Interval	3236	10436	7200
Checkpoint	1919	2759	840
Checkpoint	3719	4559	840
Checkpoint	5459	6299	840
Checkpoint	7259	8099	840
Checkpoint	9059	9899	840

Chapter 6 SUT, Driver and Communications Definition

6.1 RTE Description

If the RTE is commercially available, then its inputs must be specified. Otherwise, a description must be supplied of that input (e.g., scripts) to the RTE had been used. The RTE input parameters, code fragments, functions, et cetera used to generate each transaction input filed must be disclosed.

The RTE used is Microsoft BenchCraft and is commercially available. The RTE input parameters are listed in Appendix C – Tunable Parameters.

6.2 Emulated Components

It must be demonstrated that the functionality and performance of the components being used in the Driver System are equivalent to that of the priced system.

No components were emulated.

6.3 Functional Diagram

A complete functional diagram of the hardware and software of the benchmark configuration including the driver must be provided. the sponsor must list all hardware and software functionality of the driver and its interface to the SUT.

Functional diagrams of the measured and priced systems are included in the "General Items" section at the beginning of this report.

6.4 Networks

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

The "General Items" section includes diagrams of the network configurations of the benchmark and configured systems, and represent the driver connected via LAN.

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

100base T (100Mbit/sec) network was used between the RTEs and the clients, a 2GBps VIA/SAN connection was used between the clients and the database server.

6.5 Operator Intervention

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

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

Chapter 7 Pricing

7.1 System Pricing

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

The total 5 year price of the entire configuration must be reported, including: hardware, software, maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

The details of the hardware, software and maintenance components of this system are reported in the front of this report as part of the executive summary.

All 3rd party quotations are included at the end of this report in Appendix E.

7.2 General Availability, Throughput and Price Performance

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

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

Table 9: Throughput, Price Performance and Availability

Maximum qualified throughput:	87741.45 tpmC
Price per tpmC:	$5.03
Availability:	February 12, 2003

7.3 Country Specific Pricing

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

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

7.4 Usage Pricing

For any usage pricing, the sponsor must disclose: Usage level at which the component was priced, a statement of the company policy allowing such pricing.

The component pricing based on usage is shown below:

- 4 Microsoft SQL Server 2000 Enterprise Edition 64-bit per-processor licenses.
- 1 Microsoft Windows .NET Enterprise Server 2003.
- 5 Microsoft Windows 2000 Server licenses.
- 1 Microsoft Visual C++ 32bit Edition.
- 3 year support for hardware components

Chapter 8 Audit

8.1 Auditor's Information

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.

The test methodology and results of this TPC Benchmark C were audited by:

Infosizing
1373 North Franklin Street
Colorado Springs, CO 80903-2527
U.S.A.
(719) 473-7555
Fax: (719) 473-7554

The auditor was Francois Raab.
Requests for this Full Disclosure Report (FDR) should sent to:

Hewlett-Packard Company
ISS
10955 Tantau Avenue
Cupertino, CA 95014-0770 USA

A copy of the attestation letter received from the auditor follows:

Sponsor: Mr. Gunter Zink
System Performance Engineer
Hewlett-Packard
WSO
14335 NE 24th St, Suite B-201
Bellevue, WA 98007

December 12, 2002

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

Platform: hp server rx5670 c/s
Operating system: Microsoft Windows .NET Advanced Server
Database Manager: Microsoft SQL Server 2000 Enterprise Edition 64-bit
Transaction Manager: Microsoft COM+ Transaction Monitor

The results were:

CPU's Speed	Memory	Disks	NewOrder 90% Response Time	tpmC
Server: **hp server rx5670**				
4 x Itanium II (1 GHz)	48 GB (3 MB cache/cpu)	280 x 18 GB 15Krpm 1 x 36 GB 15Krpm	0.67 Seconds	87,741.45
Five (5) Clients: **hp Proliant DL360** (Specification for each)				
2 x Pentium III (1.4 GHz)	1 GB (512 KB cache/cpu)	1 x 18 GB	n/a	n/a

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

The following verification items were given special attention:

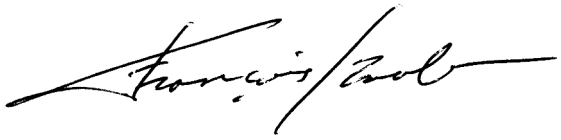
- The transactions were correctly implemented

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

Additional Audit Notes:

None.

Respectfully Yours,



François Raab, President

Appendix A Source Code

Isapi_dll/src/tpcc.def

```
LIBRARY TPCC.DLL

EXPORTS

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

Isapi_dll/src/tpcc.h

```
/*      FILE:                   TPCC.H
 *                              Microsoft TPC-C Kit Ver. 4.20.000
 *                              Copyright Microsoft, 1999   All
Rights Reserved
 *                              Version 4.10.000 audited by Richard
Gimarc, Performance Metrics, 3/17/99
 *
 *      PURPOSE:    Header file for ISAPI TPCC.DLL, defines structures
and functions used in the isapi tpcc.dll.
 *
 *
//VERSION RESOURCE DEFINES
#define _APS_NEXT_RESOURCE_VALUE
        101
#define _APS_NEXT_COMMAND_VALUE
        40001
#define _APS_NEXT_CONTROL_VALUE
        1000
#define _APS_NEXT_SYMED_VALUE
        101

#define TP_MAX_RETRIES
                50

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

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


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

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

        CTPCC_BASE *pTxn;

} CLIENTDATA, *PCLIENTDATA;


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

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

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


        enum WEBERROR
        {
                NO_ERR,
                ERR_COMMAND_UNDEFINED,
                ERR_D_ID_INVALID,
                ERR_DELIVERY_CARRIER_ID_RANGE,
                ERR_DELIVERY_CARRIER_INVALID,
                ERR_DELIVERY_MISSING_OCD_KEY,
                ERR_DELIVERY_THREAD_FAILED,
                ERR_GETPROCADDR_FAILED,
                ERR_HTML_ILL_FORMED,
                ERR_INVALID_SYNC_CONNECTION,
                ERR_INVALID_TERMID,
                ERR_LOADDLL_FAILED,
                ERR_MAX_CONNECTIONS_EXCEEDED,
                ERR_MEM_ALLOC_FAILED,
                ERR_MISSING_REGISTRY_ENTRIES,
                ERR_NEWORDER_CUSTOMER_INVALID,
                ERR_NEWORDER_CUSTOMER_KEY,
                ERR_NEWORDER_DISTRICT_INVALID,
                ERR_NEWORDER_FORM_MISSING_DID,
                ERR_NEWORDER_ITEMID_INVALID,
                ERR_NEWORDER_ITEMID_RANGE,
                ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
                ERR_NEWORDER_MISSING_IID_KEY,
                ERR_NEWORDER_MISSING_QTY_KEY,
                ERR_NEWORDER_MISSING_SUPPW_KEY,
                ERR_NEWORDER_NOITEMS_ENTERED,
                ERR_NEWORDER_QTY_INVALID,
                ERR_NEWORDER_QTY_RANGE,
                ERR_NEWORDER_QTY_WITHOUT_SUPPW,
                ERR_NEWORDER_SUPPW_INVALID,
                ERR_NO_SERVER_SPECIFIED,
                ERR_ORDERSTATUS_CID_AND_CLT,
                ERR_ORDERSTATUS_CID_INVALID,
```

```
                ERR_ORDERSTATUS_CLT_RANGE,
                ERR_ORDERSTATUS_DID_INVALID,
                ERR_ORDERSTATUS_MISSING_CID_CLT,

                ERR_ORDERSTATUS_MISSING_CID_KEY,
                ERR_ORDERSTATUS_MISSING_CLT_KEY,
                ERR_ORDERSTATUS_MISSING_DID_KEY,
                ERR_PAYMENT_CDI_INVALID,
                ERR_PAYMENT_CID_AND_CLT,
                ERR_PAYMENT_CUSTOMER_INVALID,
                ERR_PAYMENT_CWI_INVALID,
                ERR_PAYMENT_DISTRICT_INVALID,
                ERR_PAYMENT_HAM_INVALID,
                ERR_PAYMENT_HAM_RANGE,
                ERR_PAYMENT_LAST_NAME_TO_LONG,
                ERR_PAYMENT_MISSING_CDI_KEY,
                ERR_PAYMENT_MISSING_CID_CLT,
                ERR_PAYMENT_MISSING_CID_KEY,
                ERR_PAYMENT_MISSING_CLT,
                ERR_PAYMENT_MISSING_CLT_KEY,
                ERR_PAYMENT_MISSING_CWI_KEY,
                ERR_PAYMENT_MISSING_DID_KEY,
                ERR_PAYMENT_MISSING_HAM_KEY,
                ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
                ERR_STOCKLEVEL_THRESHOLD_INVALID,
                ERR_STOCKLEVEL_THRESHOLD_RANGE,
                ERR_VERSION_MISMATCH,
                ERR_W_ID_INVALID
        };

class CWEBCLNT_ERR : public CBaseErr
{
        public:

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

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

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

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

};

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

//function prototypes

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

Isapi_dll/src/tpcc.rc

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

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

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

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

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

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

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

#endif    // !_MAC

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

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

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

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

#endif    // APSTUDIO_INVOKED


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

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


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

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
    IDD_DIALOG1, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 179
        BOTTOMMARGIN, 88
        TOPMARGIN, 7
    END
END
#endif    // APSTUDIO_INVOKED

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



#ifndef APSTUDIO_INVOKED
//////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//
//////////////////////////////////////////////////////////////////////////////
#endif    // not APSTUDIO_INVOKED
```

Isapi_dll/src/tpcc.cpp

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

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

```
#include <sqltypes.h>

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

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

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

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

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

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


#define     LEN_ERR_STRING          256

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

char
        szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

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

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

static      CRITICAL_SECTION                TermCriticalSection;

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

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

// For deferred Delivery txns:

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

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

// configuration settings from registry
TPCCREGISTRYDATA                Reg;


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

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

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

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

            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
#endif          StopCAP();


                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 );
                                delete e;

                                // log the error txn
                                txnDeliRec.TxnStatus = e-
>ErrorType();
                                if (txnDelilog != NULL)
                                        txnDelilog-
>WriteToLog(&txnDeliRec);}
                        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)

        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></PRE><HR>"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"
                        "</FORM></HTML>"
                        , pStockLevelData->threshold,
pStockLevelData->low_stock);
        }
}

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

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

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

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

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

```
                    "<INPUT TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"
                    "<INPUT TYPE=\"hidden\" NAME=\"TERMID\"
VALUE=\"%d\">"
                    "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\"
VALUE=\"%d\">"
                    "<PRE><font face=\"Courier\">
New Order<BR>"
                    , bValid ? 0 : ERR_BAD_ITEM_ID,
NEW_ORDER_FORM, iTermId, Term.pClientData[iTermId].iSyncId);

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

                    strcpy( szForm+c,
                              "District: <INPUT NAME=\"DID*\"
SIZE=1>                    Date:<BR>"
                              "Customer:  <INPUT
NAME=\"CID*\" SIZE=4>   Name:              Credit:     %Disc:<BR>"
                              "Order Number:          Number of
Lines:          W_tax:        D_tax:<BR> <BR>"
                              " Supp_W  Item_Id  Item Name
Qty  Stock  B/G  Price    Amount<BR>"
                              "  <INPUT NAME=\"SP00*\"
SIZE=4>  <INPUT NAME=\"IID00*\" SIZE=6>                    <INPUT
NAME=\"Qty00*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP01*\"
SIZE=4>  <INPUT NAME=\"IID01*\" SIZE=6>                    <INPUT
NAME=\"Qty01*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP02*\"
SIZE=4>  <INPUT NAME=\"IID02*\" SIZE=6>                    <INPUT
NAME=\"Qty02*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP03*\"
SIZE=4>  <INPUT NAME=\"IID03*\" SIZE=6>                    <INPUT
NAME=\"Qty03*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP04*\"
SIZE=4>  <INPUT NAME=\"IID04*\" SIZE=6>                    <INPUT
NAME=\"Qty04*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP05*\"
SIZE=4>  <INPUT NAME=\"IID05*\" SIZE=6>                    <INPUT
NAME=\"Qty05*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP06*\"
SIZE=4>  <INPUT NAME=\"IID06*\" SIZE=6>                    <INPUT
NAME=\"Qty06*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP07*\"
SIZE=4>  <INPUT NAME=\"IID07*\" SIZE=6>                    <INPUT
NAME=\"Qty07*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP08*\"
SIZE=4>  <INPUT NAME=\"IID08*\" SIZE=6>                    <INPUT
NAME=\"Qty08*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP09*\"
SIZE=4>  <INPUT NAME=\"IID09*\" SIZE=6>                    <INPUT
NAME=\"Qty09*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP10*\"
SIZE=4>  <INPUT NAME=\"IID10*\" SIZE=6>                    <INPUT
NAME=\"Qty10*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP11*\"
SIZE=4>  <INPUT NAME=\"IID11*\" SIZE=6>                    <INPUT
NAME=\"Qty11*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP12*\"
SIZE=4>  <INPUT NAME=\"IID12*\" SIZE=6>                    <INPUT
NAME=\"Qty12*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP13*\"
SIZE=4>  <INPUT NAME=\"IID13*\" SIZE=6>                    <INPUT
NAME=\"Qty13*\" SIZE=1><BR>"
                              "  <INPUT NAME=\"SP14*\"
SIZE=4>  <INPUT NAME=\"IID14*\" SIZE=6>                    <INPUT
NAME=\"Qty14*\" SIZE=1><BR>"
                              "Execution Status:
Total:<BR>"
                              "</font></PRE><HR>"
                              "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">"
                              "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"
                              "</FORM></HTML>"
                              );
          }
          else
          {
                    c += wsprintf(szForm+c, "Warehouse: %4.4d
District: %2.2d                    Date: ",
                              pNewOrderData->w_id,
                              pNewOrderData->d_id);

                    if ( bValid )
                    {
                              c += wsprintf(szForm+c, "%2.2d-
%2.2d-%4.4d %2.2d:%2.2d:%2.2d",
                                        pNewOrderData-
>o_entry_d.day,
                                        pNewOrderData-
>o_entry_d.month,
                                        pNewOrderData-
>o_entry_d.year,
                                        pNewOrderData-
>o_entry_d.hour,
                                        pNewOrderData-
>o_entry_d.minute,
                                        pNewOrderData-
>o_entry_d.second);
                    }
                    c += wsprintf(szForm+c, "<BR>Customer:
%4.4d   Name: %-16s   Credit: %-2s   ",pNewOrderData->c_id,
pNewOrderData->c_last, pNewOrderData->c_credit);

                    if ( bValid )
                    {
                              c += sprintf(szForm+c,

          "%%Disc: %5.2f          <BR>"

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

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

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

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

          pNewOrderData->OL[i].ol_supply_w_id,

          pNewOrderData->OL[i].ol_i_id,

          pNewOrderData->OL[i].ol_i_name,

          pNewOrderData->OL[i].ol_quantity,

          pNewOrderData->OL[i].ol_stock,

          pNewOrderData->OL[i].ol_brand_generic,

          pNewOrderData->OL[i].ol_i_price,

          pNewOrderData->OL[i].ol_amount );
                              }
                    }
                    else
                    {
                              c += wsprintf(szForm+c,
                                        "%Disc:<BR>"
```

```
                                        "Order Number: %8.8d
Number of Lines:        W_tax:      D_tax:<BR> <BR>"
                                        " Supp_W  Item_Id
Item Name           Qty  Stock  B/G  Price    Amount<BR>"
                                        , pNewOrderData-
>o_id);

                                i = 0;
                        }

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

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

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

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

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

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

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

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

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

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

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

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

                pPaymentData->c_data, pPaymentData->c_data+50,
pPaymentData->c_data+100, pPaymentData->c_data+150 );
                        else
                                strcpy(szForm+c, "Cust-Data: <BR>
<BR> <BR> <BR>");

                        strcat(szForm,                  "
<BR></font></PRE><HR>"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-
Status..\">"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-
Level..\">"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Exit..\">"
                "</BODY></FORM></HTML>");

}

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

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

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

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

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

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

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

```
                                                pOrderStatusData->OL[i].ol_delivery_d.month,
                                                pOrderStatusData->OL[i].ol_delivery_d.year);
                        }

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

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

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

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

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

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

                        , pDeliveryData->o_carrier_id,
                        (pDeliveryData->exec_status_code == eOK) ? "Delivery has been queued." : "Delivery Post Failed     "
                        );
        }
}
/* FUNCTION: ProcessNewOrderForm
 *
 * PURPOSE: This function gets and validates the input data from the new order form
 *                                                filling in the required input variables. it then calls the SQLNewOrder
 *                                                transaction, constructs the output form and writes it back to client
 *                                                browser.
 */

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

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

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

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

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

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

void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char *szBuffer)
{
```

```
	PPAYMENT_DATA		pPayment;

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

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

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

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

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

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

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

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

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

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

	PDELIVERY_DATA		pDelivery;

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

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

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

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

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

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

	PSTOCK_LEVEL_DATA	pStockLevel;

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

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

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

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

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

/* FUNCTION: GetNewOrderData
```

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

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

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

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

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

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

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

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

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

          pNewOrderData->o_ol_cnt = items;
}

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

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

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

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

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

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

                    _strupr( szTmp );
```

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

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

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

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

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

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

}

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

BOOL IsNumeric(char *ptr)
{
        if ( *ptr == 0 )
                return FALSE;

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

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

BOOL IsDecimal(char *ptr)
{
        char *dotptr;
        BOOL  bValid;

        if ( *ptr == 0 )
                return FALSE;

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

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

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

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

Isapi_dll/src/resource.h

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

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

common/src/ReadRegistry.h

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

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

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

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

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );
```

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;
}
```

common\src\error.h

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

#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
};
```

common\src\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 STATE_LEN               2
#define ADDRESS_LEN             20
#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."
};

typedef struct
// transaction structures
{
        // 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;
```

common\src\txn_base.h

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

#pragma once

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

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

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

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

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

db_dblib_dll\src\tpcc_dlib.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

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_SCH_ORDER,             "No
orders found for customer."
                        },
                        { 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
{
        const BYTE      *pData;
        LOGINREC        *login;

        // 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
        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) ||
(++iTryCount > iMaxRetries))
                                        throw;

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

void CTPCC_DBLIB::NewOrder()
{
                int                                     i;
                DBINT                   commit_flag;
                DBDATETIMEdatetime;
                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, 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, 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, pData, dbdatlen(m_dbproc,1), SQLFLT8, (BYTE
*)&m_txn.NewOrder.w_tax, 8);
                                if (pData=dbdata(m_dbproc, 2))


                                        dbconvert(m_dbproc,
SQLNUMERIC, 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, 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) ||
(++iTryCount > iMaxRetries))
                                throw;

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


void CTPCC_DBLIB::Payment()
{
        DBDATETIMEdatetime;
        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, pData, dbdatlen(m_dbproc,24), SQLFLT8, (BYTE
*)&m_txn.Payment.c_credit_lim, 8);
					if(pData=dbdata(m_dbproc, 25))
						dbconvert(m_dbproc,
SQLNUMERIC, pData, dbdatlen(m_dbproc,25), SQLFLT8, (BYTE
*)&m_txn.Payment.c_discount, 8);
					if(pData=dbdata(m_dbproc, 26))
						dbconvert(m_dbproc,
SQLNUMERIC, 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) ||
(++iTryCount > iMaxRetries))
						throw;

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


void CTPCC_DBLIB::OrderStatus()
{
		int				i;
		DBDATETIMEdatetime;
		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, 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, 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) ||
(++iTryCount > iMaxRetries))
					throw;

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


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) ||
(++iTryCount > iMaxRetries))
                                                throw;

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

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;
}
```

tm_com_dll\src\tpcc_com.h

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

#pragma once

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

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

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

                public:
                                // use this interface for genuine COM errors
                                CCOMERR( HRESULT hr )
                                {
                                        m_hr = hr;
                                        m_iErrorType = 0;
                                        m_iError = 0;
                                }
                                // use this interface to impersonate a non-COM
error type
                                CCOMERR( int iErrorType, int iError )
                                {
                                        m_iErrorType = iErrorType;
                                        m_iError = iError;
                                        m_hr = S_OK;
                                }

                                int                     m_hr;
                                int                     m_iErrorType;
                                int                     m_iError;

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

                                int ErrorNum() {return m_hr;}

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

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

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

```
		struct COM_DATA
		{
			int ErrorType;
			int error;
			union
			{
				NEW_ORDER_DATA
		NewOrder;
				PAYMENT_DATA
		Payment;
				DELIVERY_DATA
		Delivery;
				STOCK_LEVEL_DATA
	StockLevel;
				ORDER_STATUS_DATA
	OrderStatus;
			} u;
		} *m_pTxn;

	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);
```

tm_com_dll\src\tpcc_com.cpp

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

// needed for CoinitializeEx
#define _WIN32_WINNT 0x0400

#include <windows.h>

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

#include "..\..\common\src\trans.h"		//tpckit transaction
header contains defination 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_ps\src\tpcc_com_ps.def

```
LIBRARY      "tpcc_com_ps"

DESCRIPTION  'Proxy/Stub DLL'

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

tpcc_com_ps\src\tpcc_com_ps.h

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

/* this ALWAYS GENERATED file contains the definitions for the interfaces */


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


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

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

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

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

```
#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

/* Forward Declarations */

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


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

#ifdef __cplusplus
extern "C"{
#endif

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

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



extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

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


EXTERN_C const IID IID_ITPCC;

#if defined(__cplusplus) && !defined(CINTERFACE)

    MIDL_INTERFACE("FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
    ITPCC : public IUnknown
    {
    public:
        virtual HRESULT __stdcall NewOrder(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

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

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

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

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

        virtual HRESULT __stdcall CallSetComplete( void) = 0;

    };

#else           /* C style interface */

    typedef struct ITPCCVtbl
    {
        BEGIN_INTERFACE

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

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

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

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

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

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

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

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

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

        END_INTERFACE
    } ITPCCVtbl;

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



#ifdef COBJMACROS


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

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

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


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

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

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

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

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

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

#endif /* COBJMACROS */
```

```
#endif          /* C style interface */


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


void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


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


void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


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


void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


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


void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


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


void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC __RPC_FAR * This);


void __RPC_STUB ITPCC_CallSetComplete_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);



#endif          /* __ITPCC_INTERFACE_DEFINED__ */


/* Additional Prototypes for ALL interfaces */

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

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

tpcc_com_ps\src\tpcc_com_ps.idl

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

// Forward declare all types defined
interface ITPCC;
import "oaidl.idl";
import "ocidl.idl";

            [
                    object,
                    oleautomation,
                    uuid(FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B),
                    helpstring("ITPCC Interface"),
                    pointer_default(unique)
            ]
            interface ITPCC : IUnknown
            {
    HRESULT _stdcall NewOrder
            (
            [in] VARIANT  txn_in,
            [out] VARIANT *txn_out
            );
    HRESULT _stdcall Payment
            (
            [in] VARIANT txn_in,
            [out] VARIANT *txn_out
            );
```

```
    HRESULT _stdcall Delivery
            (
            [in] VARIANT txn_in,
            [out] VARIANT *txn_out
            );
    HRESULT _stdcall StockLevel
            (
            [in] VARIANT txn_in,
            [out] VARIANT *txn_out
            );
    HRESULT _stdcall OrderStatus
            (
            [in] VARIANT txn_in,
            [out] VARIANT *txn_out
            );
    HRESULT _stdcall CallSetComplete
            (
            );
};          // interface ITPCC
```

tpcc_com_ps\src\tpcc_com_ps_i.c

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

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

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


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

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

#ifdef __cplusplus
extern "C"{
#endif


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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED
#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



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


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

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

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


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

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

#ifdef __cplusplus
extern "C"{
#endif


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

#ifdef _MIDL_USE_GUIDDEF_
```

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



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

tpcc_com_ps\src\tpcc_com_ps_p.c

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

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


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

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


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


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


#include "tpcc_com_ps.h"

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

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

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


extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;


/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,

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


/* Object interface: IUnknown, ver. 0.0,

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


/* Object interface: ITPCC, ver. 0.0,

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


extern const MIDL_STUB_DESC Object_StubDesc;


extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

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

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

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

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

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

extern const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ];

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

#pragma data_seg(".rdata")

static const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ] =
        {

            {
            VARIANT_UserSize
            ,VARIANT_UserMarshal
            ,VARIANT_UserUnmarshal
            ,VARIANT_UserFree
            }

        };


#if !defined(__RPC_WIN32__)
#error  Invalid build platform for this stub.
#endif

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this stub because it uses
these features:
#error   -Oif or -Oicf, [wire_marshal] or [user_marshal] attribute.
#error However, your C/C++ compilation flags indicate you intend to run this
app on earlier systems.
#error This app will die there with the RPC_X_WRONG_STUB_VERSION
error.
#endif


static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
    {
        0,
        {

                /* Procedure NewOrder */

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

                /* Parameter txn_in */

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

                /* Parameter txn_out */

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

```
#endif
#else
                                        NdrFcShort( 0x18 ),     /* PPC
Stack size/offset = 24 */
#endif
#else
                                        NdrFcShort( 0x18 ),     /* Alpha
Stack size/offset = 24 */
#endif
/* 26 */        NdrFcShort( 0x3da ),    /* Type Offset=986 */

                /* Return value */

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

                /* Procedure Payment */

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

                /* Parameter txn_in */

/* 50 */        NdrFcShort( 0x8b ),     /* Flags:  must size, must free, in,
by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 52 */        NdrFcShort( 0x4 ),      /* x86 Stack size/offset = 4 */
#else
                                        NdrFcShort( 0x8 ),      /*  MIPS
Stack size/offset = 8 */
#endif
#else
                                        NdrFcShort( 0x8 ),      /* PPC
Stack size/offset = 8 */
#endif
#else
                                        NdrFcShort( 0x8 ),      /* Alpha
Stack size/offset = 8 */
#endif
/* 54 */        NdrFcShort( 0x3c8 ),    /* Type Offset=968 */

                /* Parameter txn_out */

/* 56 */        NdrFcShort( 0x4113 ),   /* Flags:  must size, must free, out,
simple ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 58 */        NdrFcShort( 0x14 ),     /* x86 Stack size/offset = 20 */
#else
                                        NdrFcShort( 0x18 ),     /*  MIPS
Stack size/offset = 24 */
#endif
#else
                                        NdrFcShort( 0x18 ),     /* PPC
Stack size/offset = 24 */
#endif
#else
                                        NdrFcShort( 0x18 ),     /* Alpha
Stack size/offset = 24 */
#endif
/* 60 */        NdrFcShort( 0x3da ),    /* Type Offset=986 */

                /* Return value */
/* 62 */        NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 64 */        NdrFcShort( 0x18 ),     /* x86 Stack size/offset = 24 */
#else
                                        NdrFcShort( 0x1c ),     /*  MIPS
Stack size/offset = 28 */
#endif
#else
                                        NdrFcShort( 0x1c ),     /* PPC
Stack size/offset = 28 */
#endif
#else
                                        NdrFcShort( 0x20 ),     /* Alpha
Stack size/offset = 32 */
#endif
/* 66 */        0x8,                    /* FC_LONG */
                                        0x0,                    /* 0 */

                /* Procedure Delivery */

/* 68 */        0x33,                   /* FC_AUTO_HANDLE */
                                        0x6c,                   /* Old
Flags:  object, Oi2 */
/* 70 */        NdrFcLong( 0x0 ),       /* 0 */
/* 74 */        NdrFcShort( 0x5 ),      /* 5 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 76 */        NdrFcShort( 0x1c ),     /* x86 Stack size/offset = 28 */
#else
                                        NdrFcShort( 0x20 ),     /*  MIPS
Stack size/offset = 32 */
#endif
#else
                                        NdrFcShort( 0x20 ),     /* PPC
Stack size/offset = 32 */
#endif
#else
                                        NdrFcShort( 0x28 ),     /* Alpha
Stack size/offset = 40 */
#endif
/* 78 */        NdrFcShort( 0x0 ),      /* 0 */
/* 80 */        NdrFcShort( 0x8 ),      /* 8 */
/* 82 */        0x7,                    /* Oi2 Flags:  srv must size, clt must
size, has return, */
                                        0x3,                    /* 3 */

                /* Parameter txn_in */
```

```
/* 84 */       NdrFcShort( 0x8b ),       /* Flags:  must size, must free, in,
by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 86 */       NdrFcShort( 0x4 ),        /* x86 Stack size/offset = 4 */
#else
                                         NdrFcShort( 0x8 ),        /*  MIPS
Stack size/offset = 8 */
#endif
#else
                                         NdrFcShort( 0x8 ),        /* PPC
Stack size/offset = 8 */
#endif
#else
                                         NdrFcShort( 0x8 ),        /* Alpha
Stack size/offset = 8 */
#endif
/* 88 */       NdrFcShort( 0x3c8 ),      /* Type Offset=968 */

                /* Parameter txn_out */

/* 90 */       NdrFcShort( 0x4113 ),     /* Flags:  must size, must free, out,
simple ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 92 */       NdrFcShort( 0x14 ),       /* x86 Stack size/offset = 20 */
#else
                                         NdrFcShort( 0x18 ),       /*  MIPS
Stack size/offset = 24 */
#endif
#else
                                         NdrFcShort( 0x18 ),       /* PPC
Stack size/offset = 24 */
#endif
#else
                                         NdrFcShort( 0x18 ),       /* Alpha
Stack size/offset = 24 */
#endif
/* 94 */       NdrFcShort( 0x3da ),      /* Type Offset=986 */

                /* Return value */

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

                /* Procedure StockLevel */

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

                /* Parameter txn_in */

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

                /* Parameter txn_out */

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

                /* Return value */

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

                /* Procedure OrderStatus */
```

```
/* 136 */      0x33,                /* FC_AUTO_HANDLE */
                                    0x6c,                /* Old
Flags:  object, Oi2 */
/* 138 */      NdrFcLong( 0x0 ),    /* 0 */
/* 142 */      NdrFcShort( 0x7 ),   /* 7 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 144 */      NdrFcShort( 0x1c ),  /* x86 Stack size/offset = 28 */
#else
                                    NdrFcShort( 0x20 ),  /*  MIPS
Stack size/offset = 32 */
#endif
#else
                                    NdrFcShort( 0x20 ),  /* PPC
Stack size/offset = 32 */
#endif
#else
                                    NdrFcShort( 0x28 ),  /* Alpha
Stack size/offset = 40 */
#endif
/* 146 */      NdrFcShort( 0x0 ),   /* 0 */
/* 148 */      NdrFcShort( 0x8 ),   /* 8 */
/* 150 */      0x7,                 /* Oi2 Flags:  srv must size, clt must
size, has return, */
                                    0x3,                 /* 3 */

            /* Parameter txn_in */

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

            /* Parameter txn_out */
```

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

            /* Return value */

/* 164 */      NdrFcShort( 0x70 ),  /* Flags:  out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 166 */      NdrFcShort( 0x18 ),  /* x86 Stack size/offset = 24 */
#else
                                    NdrFcShort( 0x1c ),  /*  MIPS
Stack size/offset = 28 */
#endif
#else
                                    NdrFcShort( 0x1c ),  /* PPC
Stack size/offset = 28 */
#endif
#else
                                    NdrFcShort( 0x20 ),  /* Alpha
Stack size/offset = 32 */
#endif
/* 168 */      0x8,                 /* FC_LONG */
                                    0x0,                 /* 0 */

            /* Procedure CallSetComplete */

/* 170 */      0x33,                /* FC_AUTO_HANDLE */
                                    0x6c,                /* Old
Flags:  object, Oi2 */
/* 172 */      NdrFcLong( 0x0 ),    /* 0 */
/* 176 */      NdrFcShort( 0x8 ),   /* 8 */
#ifndef _ALPHA_
/* 178 */      NdrFcShort( 0x8 ),   /* x86, MIPS, PPC Stack size/offset
= 8 */
#else
```

```
                                    NdrFcShort( 0x10 ),  /* Alpha
Stack size/offset = 16 */
#endif
/* 180 */      NdrFcShort( 0x0 ),   /* 0 */
/* 182 */      NdrFcShort( 0x8 ),   /* 8 */
/* 184 */      0x4,                 /* Oi2 Flags:  has return, */
                                    0x1,                 /* 1 */

            /* Return value */

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

                                    0x0
        }
    };

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
    {
        0,
        {
                                    NdrFcShort( 0x0 ),   /* 0 */
/*  2 */
                                    0x12, 0x0,    /* FC_UP */
/*  4 */       NdrFcShort( 0x3b0 ), /* Offset= 944 (948) */
/*  6 */
                                    0x2b,                /*
FC_NON_ENCAPSULATED_UNION */
                                    0x9,                 /*
FC_ULONG */
/*  8 */       0x7,                 /* Corr desc: FC_USHORT */
                                    0x0,                 /*  */
/* 10 */       NdrFcShort( 0xfff8 ),/* -8 */
/* 12 */       NdrFcShort( 0x2 ),   /* Offset= 2 (14) */
/* 14 */       NdrFcShort( 0x10 ),  /* 16 */
/* 16 */       NdrFcShort( 0x2b ),  /* 43 */
/* 18 */       NdrFcLong( 0x3 ),    /* 3 */
/* 22 */       NdrFcShort( 0x8008 ),/* Simple arm type: FC_LONG */
/* 24 */       NdrFcLong( 0x11 ),   /* 17 */
/* 28 */       NdrFcShort( 0x8001 ),/* Simple arm type: FC_BYTE */
/* 30 */       NdrFcLong( 0x2 ),    /* 2 */
/* 34 */       NdrFcShort( 0x8006 ),/* Simple arm type: FC_SHORT */
/* 36 */       NdrFcLong( 0x4 ),    /* 4 */
/* 40 */       NdrFcShort( 0x800a ),/* Simple arm type: FC_FLOAT */
/* 42 */       NdrFcLong( 0x5 ),    /* 5 */
/* 46 */       NdrFcShort( 0x800c ),/* Simple arm type: FC_DOUBLE */
```

```
/* 48 */      NdrFcLong( 0xb ),          /* 11 */
/* 52 */      NdrFcShort( 0x8006 ),      /* Simple arm type: FC_SHORT */
/* 54 */      NdrFcLong( 0xa ),          /* 10 */
/* 58 */      NdrFcShort( 0x8008 ),      /* Simple arm type: FC_LONG */
/* 60 */      NdrFcLong( 0x6 ),          /* 6 */
/* 64 */      NdrFcShort( 0xd6 ),        /* Offset= 214 (278) */
/* 66 */      NdrFcLong( 0x7 ),          /* 7 */
/* 70 */      NdrFcShort( 0x800c ),      /* Simple arm type: FC_DOUBLE */
/* 72 */      NdrFcLong( 0x8 ),          /* 8 */
/* 76 */      NdrFcShort( 0xd0 ),        /* Offset= 208 (284) */
/* 78 */      NdrFcLong( 0xd ),          /* 13 */
/* 82 */      NdrFcShort( 0xe2 ),        /* Offset= 226 (308) */
/* 84 */      NdrFcLong( 0x9 ),          /* 9 */
/* 88 */      NdrFcShort( 0xee ),        /* Offset= 238 (326) */
/* 90 */      NdrFcLong( 0x2000 ),       /* 8192 */
/* 94 */      NdrFcShort( 0xfa ),        /* Offset= 250 (344) */
/* 96 */      NdrFcLong( 0x24 ),         /* 36 */
/* 100 */     NdrFcShort( 0x308 ),       /* Offset= 776 (876) */
/* 102 */     NdrFcLong( 0x4024 ),       /* 16420 */
/* 106 */     NdrFcShort( 0x302 ),       /* Offset= 770 (876) */
/* 108 */     NdrFcLong( 0x4011 ),       /* 16401 */
/* 112 */     NdrFcShort( 0x300 ),       /* Offset= 768 (880) */
/* 114 */     NdrFcLong( 0x4002 ),       /* 16386 */
/* 118 */     NdrFcShort( 0x2fe ),       /* Offset= 766 (884) */
/* 120 */     NdrFcLong( 0x4003 ),       /* 16387 */
/* 124 */     NdrFcShort( 0x2fc ),       /* Offset= 764 (888) */
/* 126 */     NdrFcLong( 0x4004 ),       /* 16388 */
/* 130 */     NdrFcShort( 0x2fa ),       /* Offset= 762 (892) */
/* 132 */     NdrFcLong( 0x4005 ),       /* 16389 */
/* 136 */     NdrFcShort( 0x2f8 ),       /* Offset= 760 (896) */
/* 138 */     NdrFcLong( 0x400b ),       /* 16395 */
/* 142 */     NdrFcShort( 0x2e6 ),       /* Offset= 742 (884) */
/* 144 */     NdrFcLong( 0x400a ),       /* 16394 */
/* 148 */     NdrFcShort( 0x2e4 ),       /* Offset= 740 (888) */
/* 150 */     NdrFcLong( 0x4006 ),       /* 16390 */
/* 154 */     NdrFcShort( 0x2ea ),       /* Offset= 746 (900) */
/* 156 */     NdrFcLong( 0x4007 ),       /* 16391 */
/* 160 */     NdrFcShort( 0x2e0 ),       /* Offset= 736 (896) */
/* 162 */     NdrFcLong( 0x4008 ),       /* 16392 */
/* 166 */     NdrFcShort( 0x2e2 ),       /* Offset= 738 (904) */
/* 168 */     NdrFcLong( 0x400d ),       /* 16397 */
/* 172 */     NdrFcShort( 0x2e0 ),       /* Offset= 736 (908) */
/* 174 */     NdrFcLong( 0x4009 ),       /* 16393 */
/* 178 */     NdrFcShort( 0x2de ),       /* Offset= 734 (912) */
/* 180 */     NdrFcLong( 0x6000 ),       /* 24576 */
/* 184 */     NdrFcShort( 0x2dc ),       /* Offset= 732 (916) */
/* 186 */     NdrFcLong( 0x400c ),       /* 16396 */
/* 190 */     NdrFcShort( 0x2da ),       /* Offset= 730 (920) */
/* 192 */     NdrFcLong( 0x10 ),         /* 16 */
/* 196 */     NdrFcShort( 0x8002 ),      /* Simple arm type: FC_CHAR */
/* 198 */     NdrFcLong( 0x12 ),         /* 18 */
/* 202 */     NdrFcShort( 0x8006 ),      /* Simple arm type: FC_SHORT */
/* 204 */     NdrFcLong( 0x11 ),         /* 17 */
/* 208 */     NdrFcShort( 0x8008 ),      /* Simple arm type: FC_LONG */
/* 210 */     NdrFcLong( 0x16 ),         /* 22 */
/* 214 */     NdrFcShort( 0x8008 ),      /* Simple arm type: FC_LONG */
/* 216 */     NdrFcLong( 0x17 ),         /* 23 */
/* 220 */     NdrFcShort( 0x8008 ),      /* Simple arm type: FC_LONG */
/* 222 */     NdrFcLong( 0xe ),          /* 14 */
/* 226 */     NdrFcShort( 0x2be ),       /* Offset= 702 (928) */
/* 228 */     NdrFcLong( 0x400e ),       /* 16398 */
/* 232 */     NdrFcShort( 0x2c4 ),       /* Offset= 708 (940) */
/* 234 */     NdrFcLong( 0x4010 ),       /* 16400 */
/* 238 */     NdrFcShort( 0x2c2 ),       /* Offset= 706 (944) */
/* 240 */     NdrFcLong( 0x4012 ),       /* 16402 */
/* 244 */     NdrFcShort( 0x280 ),       /* Offset= 640 (884) */
/* 246 */     NdrFcLong( 0x4013 ),       /* 16403 */
/* 250 */     NdrFcShort( 0x27e ),       /* Offset= 638 (888) */
/* 252 */     NdrFcLong( 0x4016 ),       /* 16406 */
/* 256 */     NdrFcShort( 0x278 ),       /* Offset= 632 (888) */
/* 258 */     NdrFcLong( 0x4017 ),       /* 16407 */
/* 262 */     NdrFcShort( 0x272 ),       /* Offset= 626 (888) */
/* 264 */     NdrFcLong( 0x0 ),          /* 0 */
/* 268 */     NdrFcShort( 0x0 ),         /* Offset= 0 (268) */
/* 270 */     NdrFcLong( 0x1 ),          /* 1 */
/* 274 */     NdrFcShort( 0x0 ),         /* Offset= 0 (274) */
/* 276 */     NdrFcShort( 0xffffffff ),  /* Offset= -1 (275) */
/* 278 */

                                         0x15,                    /*
FC_STRUCT */
                                         0x7,                     /* 7 */
/* 280 */     NdrFcShort( 0x8 ),         /* 8 */
/* 282 */     0xb,                       /* FC_HYPER */
                                         0x5b,                    /*
FC_END */
/* 284 */
                                         0x12, 0x0,    /* FC_UP */
/* 286 */     NdrFcShort( 0xc ),         /* Offset= 12 (298) */
/* 288 */
                                         0x1b,                    /*
FC_CARRAY */
                                         0x1,                     /* 1 */
/* 290 */     NdrFcShort( 0x2 ),         /* 2 */
/* 292 */     0x9,                       /* Corr desc: FC_ULONG */
                                         0x0,                     /*  */
/* 294 */     NdrFcShort( 0xfffc ),      /* -4 */
/* 296 */     0x6,                       /* FC_SHORT */
                                         0x5b,                    /*
FC_END */
/* 298 */
                                         0x17,                    /*
FC_CSTRUCT */
                                         0x3,                     /* 3 */
/* 300 */     NdrFcShort( 0x8 ),         /* 8 */
/* 302 */     NdrFcShort( 0xfffffff2 ),  /* Offset= -14 (288) */
/* 304 */     0x8,                       /* FC_LONG */
                                         0x8,                     /*
/* 306 */                                /* FC_PAD */
FC_LONG */    0x5c,                      0x5b,                    /*
FC_END */
/* 308 */
                                         0x2f,                    /* FC_IP
*/
                                         0x5a,                    /*
FC_CONSTANT_IID */
/* 310 */     NdrFcLong( 0x0 ),          /* 0 */
/* 314 */     NdrFcShort( 0x0 ),         /* 0 */
/* 316 */     NdrFcShort( 0x0 ),         /* 0 */
/* 318 */     0xc0,                      /* 192 */
                                         0x0,                     /* 0 */
/* 320 */     0x0,                       /* 0 */
                                         0x0,                     /* 0 */
/* 322 */     0x0,                       /* 0 */
                                         0x0,                     /* 0 */
/* 324 */     0x0,                       /* 0 */
                                         0x46,                    /* 70 */
/* 326 */
                                         0x2f,                    /* FC_IP
*/
                                         0x5a,                    /*
FC_CONSTANT_IID */
/* 328 */     NdrFcLong( 0x20400 ),      /* 132096 */
/* 332 */     NdrFcShort( 0x0 ),         /* 0 */
/* 334 */     NdrFcShort( 0x0 ),         /* 0 */
/* 336 */     0xc0,                      /* 192 */
                                         0x0,                     /* 0 */
/* 338 */     0x0,                       /* 0 */
                                         0x0,                     /* 0 */
/* 340 */     0x0,                       /* 0 */
                                         0x0,                     /* 0 */
/* 342 */     0x0,                       /* 0 */
                                         0x46,                    /* 70 */
/* 344 */
                                         0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 346 */     NdrFcShort( 0x2 ),         /* Offset= 2 (348) */
/* 348 */
                                         0x12, 0x0,    /* FC_UP */
/* 350 */     NdrFcShort( 0x1fc ),       /* Offset= 508 (858) */
/* 352 */
                                         0x2a,                    /*
FC_ENCAPSULATED_UNION */
                                         0x49,                    /* 73 */
/* 354 */     NdrFcShort( 0x18 ),        /* 24 */
/* 356 */     NdrFcShort( 0xa ),         /* 10 */
/* 358 */     NdrFcLong( 0x8 ),          /* 8 */
/* 362 */     NdrFcShort( 0x58 ),        /* Offset= 88 (450) */
/* 364 */     NdrFcLong( 0xd ),          /* 13 */
/* 368 */     NdrFcShort( 0x78 ),        /* Offset= 120 (488) */
```

```
/* 370 */      NdrFcLong( 0x9 ),          /* 9 */
/* 374 */      NdrFcShort( 0x94 ),        /* Offset= 148 (522) */
/* 376 */      NdrFcLong( 0xc ),          /* 12 */
/* 380 */      NdrFcShort( 0xbc ),        /* Offset= 188 (568) */
/* 382 */      NdrFcLong( 0x24 ),         /* 36 */
/* 386 */      NdrFcShort( 0x114 ),       /* Offset= 276 (662) */
/* 388 */      NdrFcLong( 0x800d ),       /* 32781 */
/* 392 */      NdrFcShort( 0x130 ),       /* Offset= 304 (696) */
/* 394 */      NdrFcLong( 0x10 ),         /* 16 */
/* 398 */      NdrFcShort( 0x148 ),       /* Offset= 328 (726) */
/* 400 */      NdrFcLong( 0x2 ),          /* 2 */
/* 404 */      NdrFcShort( 0x160 ),       /* Offset= 352 (756) */
/* 406 */      NdrFcLong( 0x3 ),          /* 3 */
/* 410 */      NdrFcShort( 0x178 ),       /* Offset= 376 (786) */
/* 412 */      NdrFcLong( 0x14 ),         /* 20 */
/* 416 */      NdrFcShort( 0x190 ),       /* Offset= 400 (816) */
/* 418 */      NdrFcShort( 0xffffffff ),  /* Offset= -1 (417) */
/* 420 */
                                          0x1b,                     /*
FC_CARRAY */
                                          0x3,                      /* 3 */
/* 422 */      NdrFcShort( 0x4 ),         /* 4 */
/* 424 */      0x19,                      /* Corr desc:  field pointer,
FC_ULONG */
                                          0x0,                      /*  */
/* 426 */      NdrFcShort( 0x0 ),         /* 0 */
/* 428 */
                                          0x4b,                     /* FC_PP
*/
                                          0x5c,                     /*
FC_PAD */
/* 430 */
                                          0x48,                     /*
FC_VARIABLE_REPEAT */
                                          0x49,                     /*
FC_FIXED_OFFSET */
/* 432 */      NdrFcShort( 0x4 ),         /* 4 */
/* 434 */      NdrFcShort( 0x0 ),         /* 0 */
/* 436 */      NdrFcShort( 0x1 ),         /* 1 */
/* 438 */      NdrFcShort( 0x0 ),         /* 0 */
/* 440 */      NdrFcShort( 0x0 ),         /* 0 */
/* 442 */      0x12, 0x0,    /* FC_UP */
/* 444 */      NdrFcShort( 0xffffff6e ),  /* Offset= -146 (298) */
/* 446 */
                                          0x5b,                     /*
FC_END */

                                          0x8,                      /*
FC_LONG */
/* 448 */      0x5c,                      /* FC_PAD */
                                          0x5b,                     /*
FC_END */

/* 450 */                                 0x16,                     /*
FC_PSTRUCT */
                                          0x3,                      /* 3 */
/* 452 */      NdrFcShort( 0x8 ),         /* 8 */
/* 454 */
                                          0x4b,                     /* FC_PP
*/
                                          0x5c,                     /*
FC_PAD */
/* 456 */
                                          0x46,                     /*
FC_NO_REPEAT */
                                          0x5c,                     /*
FC_PAD */
/* 458 */      NdrFcShort( 0x4 ),         /* 4 */
/* 460 */      NdrFcShort( 0x4 ),         /* 4 */
/* 462 */      0x11, 0x0,    /* FC_RP */
/* 464 */      NdrFcShort( 0xffffffd4 ),  /* Offset= -44 (420) */
/* 466 */
                                          0x5b,                     /*
FC_END */

                                          0x8,                      /*
FC_LONG */
/* 468 */      0x8,                       /* FC_LONG */
                                          0x5b,                     /*
FC_END */
/* 470 */

                                          0x21,                     /*
FC_BOGUS_ARRAY */
                                          0x3,                      /* 3 */
/* 472 */      NdrFcShort( 0x0 ),         /* 0 */
/* 474 */      0x19,                      /* Corr desc:  field pointer,
FC_ULONG */
                                          0x0,                      /*  */
/* 476 */      NdrFcShort( 0x0 ),         /* 0 */
/* 478 */      NdrFcLong( 0xffffffff ),   /* -1 */
/* 482 */      0x4c,                      /* FC_EMBEDDED_COMPLEX */
                                          0x0,                      /* 0 */
/* 484 */      NdrFcShort( 0xffffff50 ),  /* Offset= -176 (308) */
/* 486 */      0x5c,                      /* FC_PAD */
                                          0x5b,                     /*
FC_END */
/* 488 */

                                          0x1a,                     /*
FC_BOGUS_STRUCT */
                                          0x3,                      /* 3 */
/* 490 */      NdrFcShort( 0x8 ),         /* 8 */
/* 492 */      NdrFcShort( 0x0 ),         /* 0 */
/* 494 */      NdrFcShort( 0x6 ),         /* Offset= 6 (500) */
/* 496 */      0x8,                       /* FC_LONG */
                                          0x36,                     /*
FC_POINTER */

/* 498 */      0x5c,                      /* FC_PAD */
                                          0x5b,                     /*
FC_END */
/* 500 */
                                          0x11, 0x0,    /* FC_RP */
/* 502 */      NdrFcShort( 0xfffffffe0 ), /* Offset= -32 (470) */
/* 504 */
                                          0x21,                     /*
FC_BOGUS_ARRAY */
                                          0x3,                      /* 3 */
/* 506 */      NdrFcShort( 0x0 ),         /* 0 */
/* 508 */      0x19,                      /* Corr desc:  field pointer,
FC_ULONG */
                                          0x0,                      /*  */
/* 510 */      NdrFcShort( 0x0 ),         /* 0 */
/* 512 */      NdrFcLong( 0xffffffff ),   /* -1 */
/* 516 */      0x4c,                      /* FC_EMBEDDED_COMPLEX */
                                          0x0,                      /* 0 */
/* 518 */      NdrFcShort( 0xffffff40 ),  /* Offset= -192 (326) */
/* 520 */      0x5c,                      /* FC_PAD */
                                          0x5b,                     /*
FC_END */
/* 522 */

                                          0x1a,                     /*
FC_BOGUS_STRUCT */
                                          0x3,                      /* 3 */
/* 524 */      NdrFcShort( 0x8 ),         /* 8 */
/* 526 */      NdrFcShort( 0x0 ),         /* 0 */
/* 528 */      NdrFcShort( 0x6 ),         /* Offset= 6 (534) */
/* 530 */      0x8,                       /* FC_LONG */
                                          0x36,                     /*
FC_POINTER */
/* 532 */      0x5c,                      /* FC_PAD */
                                          0x5b,                     /*
FC_END */
/* 534 */

                                          0x11, 0x0,    /* FC_RP */
/* 536 */      NdrFcShort( 0xfffffffe0 ), /* Offset= -32 (504) */
/* 538 */
                                          0x1b,                     /*
FC_CARRAY */
                                          0x3,                      /* 3 */
/* 540 */      NdrFcShort( 0x4 ),         /* 4 */
/* 542 */      0x19,                      /* Corr desc:  field pointer,
FC_ULONG */
                                          0x0,                      /*  */
/* 544 */      NdrFcShort( 0x0 ),         /* 0 */
/* 546 */
                                          0x4b,                     /* FC_PP
*/
                                          0x5c,                     /*
FC_PAD */
/* 548 */
```

```
                                0x48,                   /*
FC_VARIABLE_REPEAT */
                                0x49,                   /*
FC_FIXED_OFFSET */
/* 550 */       NdrFcShort( 0x4 ),      /* 4 */
/* 552 */       NdrFcShort( 0x0 ),      /* 0 */
/* 554 */       NdrFcShort( 0x1 ),      /* 1 */
/* 556 */       NdrFcShort( 0x0 ),      /* 0 */
/* 558 */       NdrFcShort( 0x0 ),      /* 0 */
/* 560 */       0x12, 0x0,      /* FC_UP */
/* 562 */       NdrFcShort( 0x182 ),    /* Offset= 386 (948) */
/* 564 */
                                0x5b,                   /*
FC_END */

                                0x8,                    /*
FC_LONG */
/* 566 */       0x5c,                   /* FC_PAD */
                                0x5b,                   /*
FC_END */
/* 568 */
                                0x1a,                   /*
FC_BOGUS_STRUCT */
                                0x3,                    /* 3 */
/* 570 */       NdrFcShort( 0x8 ),      /* 8 */
/* 572 */       NdrFcShort( 0x0 ),      /* 0 */
/* 574 */       NdrFcShort( 0x6 ),      /* Offset= 6 (580) */
/* 576 */       0x8,                    /* FC_LONG */
                                0x36,                   /*
FC_POINTER */
/* 578 */       0x5c,                   /* FC_PAD */
                                0x5b,                   /*
FC_END */
/* 580 */
                                0x11, 0x0,      /* FC_RP */
/* 582 */       NdrFcShort( 0xfffffffd4 ),      /* Offset= -44 (538) */
/* 584 */
                                0x2f,                   /* FC_IP
*/
                                0x5a,                   /*
FC_CONSTANT_IID */
/* 586 */       NdrFcLong( 0x2f ),      /* 47 */
/* 590 */       NdrFcShort( 0x0 ),      /* 0 */
/* 592 */       NdrFcShort( 0x0 ),      /* 0 */
/* 594 */       0xc0,                   /* 192 */
                                0x0,                    /* 0 */
/* 596 */       0x0,                    /* 0 */
                                0x0,                    /* 0 */
/* 598 */       0x0,                    /* 0 */
                                0x0,                    /* 0 */
/* 600 */       0x0,                    /* 0 */
                                0x46,                   /* 70 */
/* 602 */
FC_CARRAY */
                                0x1b,                   /*
                                0x0,                    /* 0 */
/* 604 */       NdrFcShort( 0x1 ),      /* 1 */
/* 606 */       0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
                                0x0,                    /*  */
/* 608 */       NdrFcShort( 0x4 ),      /* 4 */
/* 610 */       0x1,                    /* FC_BYTE */
                                0x5b,                   /*
FC_END */
/* 612 */
                                0x1a,                   /*
FC_BOGUS_STRUCT */
                                0x3,                    /* 3 */
/* 614 */       NdrFcShort( 0x10 ),     /* 16 */
/* 616 */       NdrFcShort( 0x0 ),      /* 0 */
/* 618 */       NdrFcShort( 0xa ),      /* Offset= 10 (628) */
/* 620 */       0x8,                    /* FC_LONG */
                                0x8,                    /*
FC_LONG */
/* 622 */       0x4c,                   /* FC_EMBEDDED_COMPLEX */
                                0x0,                    /* 0 */
/* 624 */       NdrFcShort( 0xfffffffd8 ),      /* Offset= -40 (584) */
/* 626 */       0x36,                   /* FC_POINTER */
                                0x5b,                   /*
FC_END */
/* 628 */

                                0x12, 0x0,      /* FC_UP */
/* 630 */       NdrFcShort( 0xfffffffe4 ),      /* Offset= -28 (602) */
/* 632 */

                                0x1b,                   /*
FC_CARRAY */
                                0x3,                    /* 3 */
/* 634 */       NdrFcShort( 0x4 ),      /* 4 */
/* 636 */       0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
                                0x0,                    /*  */
/* 638 */       NdrFcShort( 0x0 ),      /* 0 */
/* 640 */
                                0x4b,                   /* FC_PP
*/
                                0x5c,                   /*
FC_PAD */
/* 642 */

                                0x48,                   /*
FC_VARIABLE_REPEAT */
                                0x49,                   /*
FC_FIXED_OFFSET */
/* 644 */       NdrFcShort( 0x4 ),      /* 4 */
/* 646 */       NdrFcShort( 0x0 ),      /* 0 */
/* 648 */       NdrFcShort( 0x1 ),      /* 1 */
/* 650 */       NdrFcShort( 0x0 ),      /* 0 */
/* 652 */       0x12, 0x0,      /* FC_UP */
/* 654 */       NdrFcShort( 0x0 ),      /* 0 */
/* 656 */       NdrFcShort( 0xfffffffd4 ),      /* Offset= -44 (612) */
/* 658 */
                                0x5b,                   /*
FC_END */

                                0x8,                    /*
FC_LONG */
/* 660 */       0x5c,                   /* FC_PAD */
                                0x5b,                   /*
FC_END */
/* 662 */
                                0x1a,                   /*
FC_BOGUS_STRUCT */
                                0x3,                    /* 3 */
/* 664 */       NdrFcShort( 0x8 ),      /* 8 */
/* 666 */       NdrFcShort( 0x0 ),      /* 0 */
/* 668 */       NdrFcShort( 0x6 ),      /* Offset= 6 (674) */
/* 670 */       0x8,                    /* FC_LONG */
                                0x36,                   /*
FC_POINTER */
/* 672 */       0x5c,                   /* FC_PAD */
                                0x5b,                   /*
FC_END */
/* 674 */
                                0x11, 0x0,      /* FC_RP */
/* 676 */       NdrFcShort( 0xfffffffd4 ),      /* Offset= -44 (632) */
/* 678 */
                                0x1d,                   /*
FC_SMFARRAY */
                                0x0,                    /* 0 */
/* 680 */       NdrFcShort( 0x8 ),      /* 8 */
/* 682 */       0x2,                    /* FC_CHAR */
                                0x5b,                   /*
FC_END */
/* 684 */
                                0x15,                   /*
FC_STRUCT */
                                0x3,                    /* 3 */
/* 686 */       NdrFcShort( 0x10 ),     /* 16 */
/* 688 */       0x8,                    /* FC_LONG */
                                0x6,                    /*
FC_SHORT */
/* 690 */       0x6,                    /* FC_SHORT */
                                0x4c,                   /*
FC_EMBEDDED_COMPLEX */
/* 692 */       0x0,                    /* 0 */
                                NdrFcShort( 0xffffffff1 ),      /*
Offset= -15 (678) */
                                0x5b,                   /*
FC_END */
/* 696 */
```

```
                                    0x1a,                   /*
FC_BOGUS_STRUCT */
                                    0x3,                    /* 3 */
/* 698 */       NdrFcShort( 0x18 ),     /* 24 */
/* 700 */       NdrFcShort( 0x0 ),      /* 0 */
/* 702 */       NdrFcShort( 0xa ),      /* Offset= 10 (712) */
/* 704 */       0x8,                    /* FC_LONG */
                                    0x36,                   /*
FC_POINTER */
/* 706 */       0x4c,                   /* FC_EMBEDDED_COMPLEX */
                                    0x0,                    /* 0 */
/* 708 */       NdrFcShort( 0xffffffe8 ),   /* Offset= -24 (684) */
/* 710 */       0x5c,                   /* FC_PAD */
                                    0x5b,                   /*
FC_END */
/* 712 */

                                    0x11, 0x0,      /* FC_RP */
/* 714 */       NdrFcShort( 0xfffffff0c ),  /* Offset= -244 (470) */
/* 716 */
                                    0x1b,                   /*
FC_CARRAY */
                                    0x0,                    /* 0 */
/* 718 */       NdrFcShort( 0x1 ),      /* 1 */
/* 720 */       0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
                                    0x0,                    /*  */
/* 722 */       NdrFcShort( 0x0 ),      /* 0 */
/* 724 */       0x1,                    /* FC_BYTE */
                                    0x5b,                   /*
FC_END */
/* 726 */

                                    0x16,                   /*
FC_PSTRUCT */
                                    0x3,                    /* 3 */
/* 728 */       NdrFcShort( 0x8 ),      /* 8 */
/* 730 */
                                    0x4b,                   /* FC_PP
*/
                                    0x5c,                   /*
FC_PAD */
/* 732 */

                                    0x46,                   /*
FC_NO_REPEAT */
                                    0x5c,                   /*
FC_PAD */
/* 734 */       NdrFcShort( 0x4 ),      /* 4 */
/* 736 */       NdrFcShort( 0x4 ),      /* 4 */
/* 738 */       0x12, 0x0,      /* FC_UP */
/* 740 */       NdrFcShort( 0xffffffe8 ),   /* Offset= -24 (716) */
/* 742 */
                                    0x5b,                   /*
FC_END */

                                    0x8,                    /*
FC_LONG */
/* 744 */       0x8,                    /* FC_LONG */
                                    0x5b,                   /*
FC_END */
/* 746 */

                                    0x1b,                   /*
FC_CARRAY */
                                    0x1,                    /* 1 */
/* 748 */       NdrFcShort( 0x2 ),      /* 2 */
/* 750 */       0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
                                    0x0,                    /*  */
/* 752 */       NdrFcShort( 0x0 ),      /* 0 */
/* 754 */       0x6,                    /* FC_SHORT */
                                    0x5b,                   /*
FC_END */
/* 756 */

                                    0x16,                   /*
FC_PSTRUCT */
                                    0x3,                    /* 3 */
/* 758 */       NdrFcShort( 0x8 ),      /* 8 */
/* 760 */
                                    0x4b,                   /* FC_PP
*/
                                    0x5c,                   /*
FC_PAD */
/* 762 */

                                    0x46,                   /*
FC_NO_REPEAT */
                                    0x5c,                   /*
FC_PAD */
/* 764 */       NdrFcShort( 0x4 ),      /* 4 */
/* 766 */       NdrFcShort( 0x4 ),      /* 4 */
/* 768 */       0x12, 0x0,      /* FC_UP */
/* 770 */       NdrFcShort( 0xffffffe8 ),   /* Offset= -24 (746) */
/* 772 */
                                    0x5b,                   /*
FC_END */

                                    0x8,                    /*
FC_LONG */
/* 774 */       0x8,                    /* FC_LONG */
                                    0x5b,                   /*
FC_END */
/* 776 */

                                    0x1b,                   /*
FC_CARRAY */
                                    0x3,                    /* 3 */
/* 778 */       NdrFcShort( 0x4 ),      /* 4 */
/* 780 */       0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
                                    0x0,                    /*  */
/* 782 */       NdrFcShort( 0x0 ),      /* 0 */
                                    0x5b,                   /*
FC_END */
/* 786 */

                                    0x16,                   /*
FC_PSTRUCT */
                                    0x3,                    /* 3 */
/* 788 */       NdrFcShort( 0x8 ),      /* 8 */
/* 790 */
                                    0x4b,                   /* FC_PP
*/
                                    0x5c,                   /*
FC_PAD */
/* 792 */

                                    0x46,                   /*
FC_NO_REPEAT */
                                    0x5c,                   /*
FC_PAD */
/* 794 */       NdrFcShort( 0x4 ),      /* 4 */
/* 796 */       NdrFcShort( 0x4 ),      /* 4 */
/* 798 */       0x12, 0x0,      /* FC_UP */
/* 800 */       NdrFcShort( 0xffffffe8 ),   /* Offset= -24 (776) */
/* 802 */
                                    0x5b,                   /*
FC_END */

                                    0x8,                    /*
FC_LONG */
/* 804 */       0x8,                    /* FC_LONG */
                                    0x5b,                   /*
FC_END */
/* 806 */

                                    0x1b,                   /*
FC_CARRAY */
                                    0x7,                    /* 7 */
/* 808 */       NdrFcShort( 0x8 ),      /* 8 */
/* 810 */       0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
                                    0x0,                    /*  */
/* 812 */       NdrFcShort( 0x0 ),      /* 0 */
/* 814 */       0xb,                    /* FC_HYPER */
                                    0x5b,                   /*
FC_END */
/* 816 */

                                    0x16,                   /*
FC_PSTRUCT */
                                    0x3,                    /* 3 */
/* 818 */       NdrFcShort( 0x8 ),      /* 8 */
/* 820 */
                                    0x4b,                   /* FC_PP
*/
                                    0x5c,                   /*
FC_PAD */
```

```
/* 822 */
                                    0x46,                   /*
FC_NO_REPEAT */
                                    0x5c,                   /*
FC_PAD */
/* 824 */      NdrFcShort( 0x4 ),       /* 4 */
/* 826 */      NdrFcShort( 0x4 ),       /* 4 */
/* 828 */      0x12, 0x0,    /* FC_UP */
/* 830 */      NdrFcShort( 0xfffffff8 ),    /* Offset= -24 (806) */
/* 832 */
                                    0x5b,                   /*
FC_END */

                                    0x8,                    /*
FC_LONG */
/* 834 */      0x8,                     /* FC_LONG */
                                    0x5b,                   /*
FC_END */
/* 836 */
                                    0x15,                   /*
FC_STRUCT */
                                    0x3,                    /* 3 */
/* 838 */      NdrFcShort( 0x8 ),       /* 8 */
/* 840 */      0x8,                     /* FC_LONG */
                                    0x8,                    /*
FC_LONG */
/* 842 */      0x5c,                    /* FC_PAD */
                                    0x5b,                   /*
FC_END */
/* 844 */
                                    0x1b,                   /*
FC_CARRAY */
                                    0x3,                    /* 3 */
/* 846 */      NdrFcShort( 0x8 ),       /* 8 */
/* 848 */      0x7,                     /* Corr desc: FC_USHORT */
                                    0x0,                    /*  */
/* 850 */      NdrFcShort( 0xffd8 ),    /* -40 */
/* 852 */      0x4c,                    /* FC_EMBEDDED_COMPLEX */
                                    0x0,                    /* 0 */
/* 854 */      NdrFcShort( 0xffffffee ),    /* Offset= -18 (836) */
/* 856 */      0x5c,                    /* FC_PAD */
                                    0x5b,                   /*
FC_END */
/* 858 */
                                    0x1a,                   /*
FC_BOGUS_STRUCT */
                                    0x3,                    /* 3 */
/* 860 */      NdrFcShort( 0x28 ),      /* 40 */
/* 862 */      NdrFcShort( 0xffffffee ),    /* Offset= -18 (844) */
/* 864 */      NdrFcShort( 0x0 ),       /* Offset= 0 (864) */
/* 866 */      0x6,                     /* FC_SHORT */
                                    0x6,                    /*
/* 868 */
FC_SHORT */0x38,                    /* FC_ALIGNM4 */
                                    0x8,                    /*
FC_LONG */
/* 870 */      0x8,                     /* FC_LONG */
                                    0x4c,                   /*
FC_EMBEDDED_COMPLEX */
/* 872 */      0x0,                     /* 0 */
                                    NdrFcShort( 0xfffffdf7 ),   /*
Offset= -521 (352) */
                                    0x5b,                   /*
FC_END */
/* 876 */
                                    0x12, 0x0,    /* FC_UP */
/* 878 */      NdrFcShort( 0xfffffef6 ),    /* Offset= -266 (612) */
/* 880 */
                                    0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 882 */      0x1,                     /* FC_BYTE */
                                    0x5c,                   /*
FC_PAD */
/* 884 */
                                    0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 886 */      0x6,                     /* FC_SHORT */
                                    0x5c,                   /*
FC_PAD */
/* 888 */
                                    0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 890 */      0x8,                     /* FC_LONG */
                                    0x5c,                   /*
FC_PAD */
/* 892 */
                                    0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 894 */      0xa,                     /* FC_FLOAT */
                                    0x5c,                   /*
FC_PAD */
/* 896 */
                                    0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 898 */      0xc,                     /* FC_DOUBLE */
                                    0x5c,                   /*
FC_PAD */
/* 900 */
                                    0x12, 0x0,    /* FC_UP */
/* 902 */      NdrFcShort( 0xfffffd90 ),    /* Offset= -624 (278) */
/* 904 */
                                    0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 906 */      NdrFcShort( 0xfffffd92 ),    /* Offset= -622 (284) */
/* 908 */
                                    0x12, 0x10,   /* FC_UP
/* 910 */
[pointer_deref] */NdrFcShort( 0xfffffda6 ),    /* Offset= -602 (308) */
/* 912 */
                                    0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 914 */      NdrFcShort( 0xfffffdb4 ),    /* Offset= -588 (326) */
/* 916 */
                                    0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 918 */      NdrFcShort( 0xfffffdc2 ),    /* Offset= -574 (344) */
/* 920 */
                                    0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 922 */      NdrFcShort( 0x2 ),       /* Offset= 2 (924) */
/* 924 */
                                    0x12, 0x0,    /* FC_UP */
/* 926 */      NdrFcShort( 0x16 ),      /* Offset= 22 (948) */
/* 928 */
                                    0x15,                   /*
FC_STRUCT */
                                    0x7,                    /* 7 */
/* 930 */      NdrFcShort( 0x10 ),      /* 16 */
/* 932 */      0x6,                     /* FC_SHORT */
                                    0x1,                    /*
FC_BYTE */
/* 934 */      0x1,                     /* FC_BYTE */
                                    0x38,                   /*
FC_ALIGNM4 */
/* 936 */      0x8,                     /* FC_LONG */
                                    0x39,                   /*
FC_ALIGNM8 */
/* 938 */      0xb,                     /* FC_HYPER */
                                    0x5b,                   /*
FC_END */
/* 940 */
                                    0x12, 0x0,    /* FC_UP */
/* 942 */      NdrFcShort( 0xfffffff2 ),    /* Offset= -14 (928) */
/* 944 */
                                    0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 946 */      0x2,                     /* FC_CHAR */
                                    0x5c,                   /*
FC_PAD */
/* 948 */
                                    0x1a,                   /*
FC_BOGUS_STRUCT */
                                    0x7,                    /* 7 */
/* 950 */      NdrFcShort( 0x20 ),      /* 32 */
/* 952 */      NdrFcShort( 0x0 ),       /* 0 */
/* 954 */      NdrFcShort( 0x0 ),       /* Offset= 0 (954) */
/* 956 */      0x8,                     /* FC_LONG */
                                    0x8,                    /*
FC_LONG */
```

```
/* 958 */      0x6,                    /* FC_SHORT */
                                       0x6,                    /*
FC_SHORT */
/* 960 */      0x6,                    /* FC_SHORT */
                                       0x6,                    /*
FC_SHORT */
/* 962 */      0x4c,                   /* FC_EMBEDDED_COMPLEX */
                                       0x0,                    /* 0 */
/* 964 */      NdrFcShort( 0xfffffc42 ),   /* Offset= -958 (6) */
/* 966 */      0x5c,                   /* FC_PAD */
                                       0x5b,                   /*
FC_END */
/* 968 */      0xb4,                   /* FC_USER_MARSHAL */
                                       0x83,                   /* 131 */
/* 970 */      NdrFcShort( 0x0 ),      /* 0 */
/* 972 */      NdrFcShort( 0x10 ),     /* 16 */
/* 974 */      NdrFcShort( 0x0 ),      /* 0 */
/* 976 */      NdrFcShort( 0xfffffc32 ),   /* Offset= -974 (2) */
/* 978 */
                                       0x11, 0x4,    /* FC_RP
[alloced_on_stack] */
/* 980 */      NdrFcShort( 0x6 ),      /* Offset= 6 (986) */
/* 982 */
                                       0x13, 0x0,    /* FC_OP */
/* 984 */      NdrFcShort( 0xfffffffdc ),  /* Offset= -36 (948) */
/* 986 */      0xb4,                   /* FC_USER_MARSHAL */
                                       0x83,                   /* 131 */
/* 988 */      NdrFcShort( 0x0 ),      /* 0 */
/* 990 */      NdrFcShort( 0x10 ),     /* 16 */
/* 992 */      NdrFcShort( 0x0 ),      /* 0 */
/* 994 */      NdrFcShort( 0xfffffff4 ),   /* Offset= -12 (982) */

                                       0x0

      }
   };

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

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

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

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

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{

    if(!_tpcc_com_ps_CHECK_IID(0))
        {
        *pIndex = 0;
        return 1;
        }

    return 0;
}

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


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


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

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


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

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


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


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


#include "tpcc_com_ps.h"

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

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

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


extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;


/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,

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


/* Object interface: IUnknown, ver. 0.0,

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


/* Object interface: ITPCC, ver. 0.0,
```

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


extern const MIDL_STUB_DESC Object_StubDesc;


extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[] =
    {
    0,
    44,
    88,
    132,
    176,
    220
    };

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

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

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

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

extern const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ];

static const MIDL_STUB_DESC Object_StubDesc =
    {
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x50002, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.3.280 */
    0,
    UserMarshalRoutines,
    0,  /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0,  /* Reserved3 */
    0,  /* Reserved4 */
    0   /* Reserved5 */
    };

#pragma data_seg(".rdata")

static const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ] =
        {

            {
            VARIANT_UserSize
            ,VARIANT_UserMarshal
            ,VARIANT_UserUnmarshal
            ,VARIANT_UserFree
            }

        };


#if !defined(__RPC_WIN64__)
#error  Invalid build platform for this stub.
#endif

static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
    {
        0,
        {

                /* Procedure NewOrder */

                                        0x33,           /*
FC_AUTO_HANDLE */
                                        0x6c,           /* Old
Flags:  object, Oi2 */
/*  2 */        NdrFcLong( 0x0 ),       /* 0 */
/*  6 */        NdrFcShort( 0x3 ),      /* 3 */
#ifndef _ALPHA_
/*  8 */        NdrFcShort( 0x38 ),     /* ia64 Stack size/offset = 56 */
#else
                                        NdrFcShort( 0x30 ),     /* axp64
Stack size/offset = 48 */
#endif
/* 10 */        NdrFcShort( 0x0 ),      /* 0 */
/* 12 */        NdrFcShort( 0x8 ),      /* 8 */
/* 14 */        0x47,                   /* Oi2 Flags:  srv must size, clt must
size, has return, has ext, */
                                        0x3,            /* 3 */
/* 16 */        0xa,                    /* 10 */
                                        0x7,            /* Ext
Flags:  new corr desc, clt corr check, srv corr check, */
/* 18 */        NdrFcShort( 0x20 ),     /* 32 */
/* 20 */        NdrFcShort( 0x20 ),     /* 32 */
/* 22 */        NdrFcShort( 0x0 ),      /* 0 */
/* 24 */        NdrFcShort( 0x0 ),      /* 0 */

                /* Parameter txn_in */

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

                /* Parameter txn_out */
```

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

                /* Return value */

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

                /* Procedure Payment */

/* 44 */        0x33,                       /* FC_AUTO_HANDLE */
                                            0x6c,                       /* Old
Flags:  object, Oi2 */
/* 46 */        NdrFcLong( 0x0 ),           /* 0 */
/* 50 */        NdrFcShort( 0x4 ),          /* 4 */
#ifndef _ALPHA_
/* 52 */        NdrFcShort( 0x38 ),         /* ia64 Stack size/offset = 56 */
#else
                                            NdrFcShort( 0x30 ),         /* axp64
Stack size/offset = 48 */
#endif
/* 54 */        NdrFcShort( 0x0 ),          /* 0 */
/* 56 */        NdrFcShort( 0x8 ),          /* 8 */
/* 58 */        0x47,                       /* Oi2 Flags:  srv must size, clt must
size, has return, has ext, */
                                            0x3,                        /* 3 */
/* 60 */        0xa,                        /* 10 */
                                            0x7,                        /* Ext
Flags:  new corr desc, clt corr check, srv corr check, */
/* 62 */        NdrFcShort( 0x20 ),         /* 32 */
/* 64 */        NdrFcShort( 0x20 ),         /* 32 */
/* 66 */        NdrFcShort( 0x0 ),          /* 0 */
/* 68 */        NdrFcShort( 0x0 ),          /* 0 */

                /* Parameter txn_in */

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

                /* Parameter txn_out */

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

                /* Return value */

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

                /* Procedure Delivery */

/* 88 */        0x33,                       /* FC_AUTO_HANDLE */
                                            0x6c,                       /* Old
Flags:  object, Oi2 */
/* 90 */        NdrFcLong( 0x0 ),           /* 0 */
/* 94 */        NdrFcShort( 0x5 ),          /* 5 */
#ifndef _ALPHA_
/* 96 */        NdrFcShort( 0x38 ),         /* ia64 Stack size/offset = 56 */
#else
                                            NdrFcShort( 0x30 ),         /* axp64
Stack size/offset = 48 */
#endif
/* 98 */        NdrFcShort( 0x0 ),          /* 0 */
/* 100 */       NdrFcShort( 0x8 ),          /* 8 */
/* 102 */       0x47,                       /* Oi2 Flags:  srv must size, clt must
size, has return, has ext, */
                                            0x3,                        /* 3 */
/* 104 */       0xa,                        /* 10 */
                                            0x7,                        /* Ext
Flags:  new corr desc, clt corr check, srv corr check, */
/* 106 */       NdrFcShort( 0x20 ),         /* 32 */
/* 108 */       NdrFcShort( 0x20 ),         /* 32 */
/* 110 */       NdrFcShort( 0x0 ),          /* 0 */
/* 112 */       NdrFcShort( 0x0 ),          /* 0 */

                /* Parameter txn_in */

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

                /* Parameter txn_out */

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

                /* Return value */

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

                /* Procedure StockLevel */

/* 132 */       0x33,                       /* FC_AUTO_HANDLE */
                                            0x6c,                       /* Old
Flags:  object, Oi2 */
/* 134 */       NdrFcLong( 0x0 ),           /* 0 */
/* 138 */       NdrFcShort( 0x6 ),          /* 6 */
#ifndef _ALPHA_
/* 140 */       NdrFcShort( 0x38 ),         /* ia64 Stack size/offset = 56 */
#else
                                            NdrFcShort( 0x30 ),         /* axp64
Stack size/offset = 48 */
#endif
/* 142 */       NdrFcShort( 0x0 ),          /* 0 */
```

```
/* 144 */      NdrFcShort( 0x8 ),          /* 8 */
/* 146 */      0x47,                       /* Oi2 Flags:  srv must size, clt must
size, has return, has ext, */
                                           0x3,                        /* 3 */
/* 148 */      0xa,                        /* 10 */
                                           0x7,                        /* Ext
Flags:  new corr desc, clt corr check, srv corr check, */
/* 150 */      NdrFcShort( 0x20 ),         /* 32 */
/* 152 */      NdrFcShort( 0x20 ),         /* 32 */
/* 154 */      NdrFcShort( 0x0 ),          /* 0 */
/* 156 */      NdrFcShort( 0x0 ),          /* 0 */

            /* Parameter txn_in */

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

            /* Parameter txn_out */

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

            /* Return value */

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

            /* Procedure OrderStatus */

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

            /* Parameter txn_in */

/* 202 */      NdrFcShort( 0x8b ),         /* Flags:  must size, must free, in,
by val, */
#ifndef _ALPHA_
/* 204 */      NdrFcShort( 0x10 ),         /* ia64 Stack size/offset = 16 */
#else
                                           NdrFcShort( 0x8 ),          /* axp64
Stack size/offset = 8 */
#endif
/* 206 */      NdrFcShort( 0x3b6 ),        /* Type Offset=950 */

            /* Parameter txn_out */

/* 208 */      NdrFcShort( 0x6113 ),       /* Flags:  must size, must free, out,
simple ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 210 */      NdrFcShort( 0x28 ),         /* ia64 Stack size/offset = 40 */
#else
                                           NdrFcShort( 0x20 ),         /* axp64
Stack size/offset = 32 */
#endif
/* 212 */      NdrFcShort( 0x3c8 ),        /* Type Offset=968 */

            /* Return value */

/* 214 */      NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 216 */      NdrFcShort( 0x30 ),         /* ia64 Stack size/offset = 48 */
#else
                                           NdrFcShort( 0x28 ),         /* axp64
Stack size/offset = 40 */
#endif
/* 218 */      0x8,                        /* FC_LONG */
                                           0x0,                        /* 0 */

            /* Procedure CallSetComplete */

/* 220 */      0x33,                       /* FC_AUTO_HANDLE */
                                           0x6c,                       /* Old
Flags:  object, Oi2 */
/* 222 */      NdrFcLong( 0x0 ),           /* 0 */
/* 226 */      NdrFcShort( 0x8 ),          /* 8 */
/* 228 */      NdrFcShort( 0x10 ),         /* ia64, axp64 Stack size/offset = 16
*/
/* 230 */      NdrFcShort( 0x0 ),          /* 0 */
/* 232 */      NdrFcShort( 0x8 ),          /* 8 */
/* 234 */      0x44,                       /* Oi2 Flags:  has return, has ext, */
                                           0x1,                        /* 1 */
/* 236 */      0xa,                        /* 10 */
                                           0x1,                        /* Ext
Flags:  new corr desc, */
/* 238 */      NdrFcShort( 0x0 ),          /* 0 */
/* 240 */      NdrFcShort( 0x0 ),          /* 0 */
/* 242 */      NdrFcShort( 0x0 ),          /* 0 */
/* 244 */      NdrFcShort( 0x0 ),          /* 0 */

            /* Return value */

/* 246 */      NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
/* 248 */      NdrFcShort( 0x8 ),          /* ia64, axp64 Stack size/offset = 8
*/
/* 250 */      0x8,                        /* FC_LONG */
                                           0x0,                        /* 0 */

                                           0x0
    }
  };

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
  {
    0,
    {
                                           NdrFcShort( 0x0 ),          /* 0 */
/*  2 */
                                           0x12, 0x0,    /* FC_UP */
/*  4 */       NdrFcShort( 0x39e ),        /* Offset= 926 (930) */
/*  6 */
                                           0x2b,                       /*
FC_NON_ENCAPSULATED_UNION */
                                           0x9,                        /*
FC_ULONG */
/*  8 */       0x7,                        /* Corr desc: FC_USHORT */
                                           0x0,                        /*  */
```

```
/* 10 */    NdrFcShort( 0xfff8 ),    /* -8 */
/* 12 */    NdrFcShort( 0x1 ),       /* Corr flags:  early, */
/* 14 */    NdrFcShort( 0x2 ),       /* Offset= 2 (16) */
/* 16 */    NdrFcShort( 0x10 ),      /* 16 */
/* 18 */    NdrFcShort( 0x2b ),      /* 43 */
/* 20 */    NdrFcLong( 0x3 ),        /* 3 */
/* 24 */    NdrFcShort( 0x8008 ),    /* Simple arm type: FC_LONG */
/* 26 */    NdrFcLong( 0x11 ),       /* 17 */
/* 30 */    NdrFcShort( 0x8001 ),    /* Simple arm type: FC_BYTE */
/* 32 */    NdrFcLong( 0x2 ),        /* 2 */
/* 36 */    NdrFcShort( 0x8006 ),    /* Simple arm type: FC_SHORT */
/* 38 */    NdrFcLong( 0x4 ),        /* 4 */
/* 42 */    NdrFcShort( 0x800a ),    /* Simple arm type: FC_FLOAT */
/* 44 */    NdrFcLong( 0x5 ),        /* 5 */
/* 48 */    NdrFcShort( 0x800c ),    /* Simple arm type: FC_DOUBLE */
/* 50 */    NdrFcLong( 0xb ),        /* 11 */
/* 54 */    NdrFcShort( 0x8006 ),    /* Simple arm type: FC_SHORT */
/* 56 */    NdrFcLong( 0xa ),        /* 10 */
/* 60 */    NdrFcShort( 0x8008 ),    /* Simple arm type: FC_LONG */
/* 62 */    NdrFcLong( 0x6 ),        /* 6 */
/* 66 */    NdrFcShort( 0xd6 ),      /* Offset= 214 (280) */
/* 68 */    NdrFcLong( 0x7 ),        /* 7 */
/* 72 */    NdrFcShort( 0x800c ),    /* Simple arm type: FC_DOUBLE */
/* 74 */    NdrFcLong( 0x8 ),        /* 8 */
/* 78 */    NdrFcShort( 0xd0 ),      /* Offset= 208 (286) */
/* 80 */    NdrFcLong( 0xd ),        /* 13 */
/* 84 */    NdrFcShort( 0xe4 ),      /* Offset= 228 (312) */
/* 86 */    NdrFcLong( 0x9 ),        /* 9 */
/* 90 */    NdrFcShort( 0xf0 ),      /* Offset= 240 (330) */
/* 92 */    NdrFcLong( 0x2000 ),     /* 8192 */
/* 96 */    NdrFcShort( 0xfc ),      /* Offset= 252 (348) */
/* 98 */    NdrFcLong( 0x24 ),       /* 36 */
/* 102 */   NdrFcShort( 0x2f4 ),     /* Offset= 756 (858) */
/* 104 */   NdrFcLong( 0x4024 ),     /* 16420 */
/* 108 */   NdrFcShort( 0x2ee ),     /* Offset= 750 (858) */
/* 110 */   NdrFcLong( 0x4011 ),     /* 16401 */
/* 114 */   NdrFcShort( 0x2ec ),     /* Offset= 748 (862) */
/* 116 */   NdrFcLong( 0x4002 ),     /* 16386 */
/* 120 */   NdrFcShort( 0x2ea ),     /* Offset= 746 (866) */
/* 122 */   NdrFcLong( 0x4003 ),     /* 16387 */
/* 126 */   NdrFcShort( 0x2e8 ),     /* Offset= 744 (870) */
/* 128 */   NdrFcLong( 0x4004 ),     /* 16388 */
/* 132 */   NdrFcShort( 0x2e6 ),     /* Offset= 742 (874) */
/* 134 */   NdrFcLong( 0x4005 ),     /* 16389 */
/* 138 */   NdrFcShort( 0x2e4 ),     /* Offset= 740 (878) */
/* 140 */   NdrFcLong( 0x400b ),     /* 16395 */
/* 144 */   NdrFcShort( 0x2d2 ),     /* Offset= 722 (866) */
/* 146 */   NdrFcLong( 0x400a ),     /* 16394 */
/* 150 */   NdrFcShort( 0x2d0 ),     /* Offset= 720 (870) */
/* 152 */   NdrFcLong( 0x4006 ),     /* 16390 */
/* 156 */   NdrFcShort( 0x2d6 ),     /* Offset= 726 (882) */
/* 158 */   NdrFcLong( 0x4007 ),     /* 16391 */
/* 162 */   NdrFcLong( 0x4008 ),     /* 16392 */
/* 166 */   NdrFcShort( 0x2dc ),     /* Offset= 716 (878) */
/* 168 */   NdrFcShort( 0x2ce ),     /* Offset= 718 (886) */
/* 170 */   NdrFcLong( 0x400d ),     /* 16397 */
/* 174 */   NdrFcShort( 0x2cc ),     /* Offset= 716 (890) */
/* 176 */   NdrFcLong( 0x4009 ),     /* 16393 */
/* 180 */   NdrFcShort( 0x2ca ),     /* Offset= 714 (894) */
/* 182 */   NdrFcLong( 0x6000 ),     /* 24576 */
/* 186 */   NdrFcShort( 0x2c8 ),     /* Offset= 712 (898) */
/* 188 */   NdrFcLong( 0x400c ),     /* 16396 */
/* 192 */   NdrFcShort( 0x2c6 ),     /* Offset= 710 (902) */
/* 194 */   NdrFcLong( 0x10 ),       /* 16 */
/* 198 */   NdrFcShort( 0x8002 ),    /* Simple arm type: FC_CHAR */
/* 200 */   NdrFcLong( 0x12 ),       /* 18 */
/* 204 */   NdrFcShort( 0x8006 ),    /* Simple arm type: FC_SHORT */
/* 206 */   NdrFcLong( 0x13 ),       /* 19 */
/* 210 */   NdrFcShort( 0x8008 ),    /* Simple arm type: FC_LONG */
/* 212 */   NdrFcLong( 0x16 ),       /* 22 */
/* 216 */   NdrFcShort( 0x8008 ),    /* Simple arm type: FC_LONG */
/* 218 */   NdrFcLong( 0x17 ),       /* 23 */
/* 222 */   NdrFcShort( 0x8008 ),    /* Simple arm type: FC_LONG */
/* 224 */   NdrFcLong( 0xe ),        /* 14 */
/* 228 */   NdrFcShort( 0x2aa ),     /* Offset= 682 (910) */
/* 230 */   NdrFcLong( 0x400e ),     /* 16398 */
/* 234 */   NdrFcShort( 0x2b0 ),     /* Offset= 688 (922) */
/* 236 */   NdrFcLong( 0x4010 ),     /* 16400 */
/* 240 */   NdrFcShort( 0x2ae ),     /* Offset= 686 (926) */
/* 242 */   NdrFcLong( 0x4012 ),     /* 16402 */
/* 246 */   NdrFcShort( 0x26c ),     /* Offset= 620 (866) */
/* 248 */   NdrFcLong( 0x4013 ),     /* 16403 */
/* 252 */   NdrFcShort( 0x26a ),     /* Offset= 618 (870) */
/* 254 */   NdrFcLong( 0x4016 ),     /* 16406 */
/* 258 */   NdrFcShort( 0x264 ),     /* Offset= 612 (870) */
/* 260 */   NdrFcLong( 0x4017 ),     /* 16407 */
/* 264 */   NdrFcShort( 0x25e ),     /* Offset= 606 (870) */
/* 266 */   NdrFcLong( 0x0 ),        /* 0 */
/* 270 */   NdrFcShort( 0x0 ),       /* Offset= 0 (270) */
/* 272 */   NdrFcLong( 0x1 ),        /* 1 */
/* 276 */   NdrFcShort( 0x0 ),       /* Offset= 0 (276) */
/* 278 */   NdrFcShort( 0xffffffff ), /* Offset= -1 (277) */
/* 280 */
                                     0x15,                    /*
FC_STRUCT */
                                     0x7,                     /* 7 */
/* 282 */   NdrFcShort( 0x8 ),       /* 8 */
/* 284 */   0xb,                     /* FC_HYPER */
                                     0x5b,                    /*
FC_END */
/* 286 */
                                     0x12, 0x0,    /* FC_UP */
/* 288 */   NdrFcShort( 0xe ),       /* Offset= 14 (302) */
/* 290 */
                                     0x1b,                    /*
FC_CARRAY */
/* 292 */   NdrFcShort( 0x2 ),       0x2,                     /* 1 */
/* 294 */   0x9,                     /* Corr desc: FC_ULONG */
                                     0x0,                     /*  */
/* 296 */   NdrFcShort( 0xfffc ),    /* -4 */
/* 298 */   NdrFcShort( 0x1 ),       /* Corr flags:  early, */
/* 300 */   0x6,                     /* FC_SHORT */
                                     0x5b,                    /*
FC_END */
/* 302 */
                                     0x17,                    /*
FC_CSTRUCT */
                                     0x3,                     /* 3 */
/* 304 */   NdrFcShort( 0x8 ),       /* 8 */
/* 306 */   NdrFcShort( 0xfffffff0 ), /* Offset= -16 (290) */
/* 308 */   0x8,                     /* FC_LONG */
                                     0x8,                     /*
FC_LONG */
/* 310 */   0x5c,                    /* FC_PAD */
                                     0x5b,                    /*
FC_END */
/* 312 */
                                     0x2f,                    /* FC_IP
*/
                                     0x5a,                    /*
FC_CONSTANT_IID */
/* 314 */   NdrFcLong( 0x0 ),        /* 0 */
/* 318 */   NdrFcShort( 0x0 ),       /* 0 */
/* 320 */   NdrFcShort( 0x0 ),       /* 0 */
/* 322 */   0xc0,                    /* 192 */
                                     0x0,                     /* 0 */
/* 324 */   0x0,                     /* 0 */
                                     0x0,                     /* 0 */
/* 326 */   0x0,                     /* 0 */
                                     0x0,                     /* 0 */
/* 328 */   0x0,                     /* 0 */
                                     0x46,                    /* 70 */
/* 330 */
                                     0x2f,                    /* FC_IP
*/
                                     0x5a,                    /*
FC_CONSTANT_IID */
/* 332 */   NdrFcLong( 0x20400 ),    /* 132096 */
/* 336 */   NdrFcShort( 0x0 ),       /* 0 */
/* 338 */   NdrFcShort( 0x0 ),       /* 0 */
/* 340 */   0xc0,                    /* 192 */
                                     0x0,                     /* 0 */
/* 342 */   0x0,                     /* 0 */
                                     0x0,                     /* 0 */
/* 344 */   0x0,                     /* 0 */
                                     0x0,                     /* 0 */
/* 346 */   0x0,                     /* 0 */
                                     0x46,                    /* 70 */
/* 348 */
```

```
                                        0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 350 */      NdrFcShort( 0x2 ),       /* Offset= 2 (352) */
/* 352 */

/* 354 */      NdrFcShort( 0x1e6 ),     /* Offset= 486 (840) */
/* 356 */

                                        0x2a,                   /*
FC_ENCAPSULATED_UNION */

                                        0x89,                   /* 137 */
/* 358 */      NdrFcShort( 0x20 ),      /* 32 */
/* 360 */      NdrFcShort( 0xa ),       /* 10 */
/* 362 */      NdrFcLong( 0x8 ),        /* 8 */
/* 366 */      NdrFcShort( 0x50 ),      /* Offset= 80 (446) */
/* 368 */      NdrFcLong( 0xd ),        /* 13 */
/* 372 */      NdrFcShort( 0x70 ),      /* Offset= 112 (484) */
/* 374 */      NdrFcLong( 0x9 ),        /* 9 */
/* 378 */      NdrFcShort( 0x90 ),      /* Offset= 144 (522) */
/* 380 */      NdrFcLong( 0xc ),        /* 12 */
/* 384 */      NdrFcShort( 0xb0 ),      /* Offset= 176 (560) */
/* 386 */      NdrFcLong( 0x24 ),       /* 36 */
/* 390 */      NdrFcShort( 0x104 ),     /* Offset= 260 (650) */
/* 392 */      NdrFcLong( 0x800d ),     /* 32781 */
/* 396 */      NdrFcShort( 0x120 ),     /* Offset= 288 (684) */
/* 398 */      NdrFcLong( 0x10 ),       /* 16 */
/* 402 */      NdrFcShort( 0x13a ),     /* Offset= 314 (716) */
/* 404 */      NdrFcLong( 0x2 ),        /* 2 */
/* 408 */      NdrFcShort( 0x150 ),     /* Offset= 336 (744) */
/* 410 */      NdrFcLong( 0x3 ),        /* 3 */
/* 414 */      NdrFcShort( 0x166 ),     /* Offset= 358 (772) */
/* 416 */      NdrFcLong( 0x14 ),       /* 20 */
/* 420 */      NdrFcShort( 0x17c ),     /* Offset= 380 (800) */
/* 422 */      NdrFcShort( 0xffffffff ),/* Offset= -1 (421) */
/* 424 */

                                        0x21,                   /*
FC_BOGUS_ARRAY */

                                        0x3,                    /* 3 */
/* 426 */      NdrFcShort( 0x0 ),       /* 0 */
/* 428 */      0x19,                    /* Corr desc:  field pointer,
FC_ULONG */

                                        0x0,                    /*  */
/* 430 */      NdrFcShort( 0x0 ),       /* 0 */
/* 432 */      NdrFcShort( 0x1 ),       /* Corr flags:  early, */
/* 434 */      NdrFcLong( 0xffffffff ), /* -1 */
/* 438 */      NdrFcShort( 0x0 ),       /* Corr flags:  */
/* 440 */

                                        0x12, 0x0,    /* FC_UP */
/* 442 */      NdrFcShort( 0xffffff74 ),/* Offset= -140 (302) */
/* 444 */      0x5c,                    /* FC_PAD */
                                        0x5b,                   /*
FC_END */
/* 446 */
```

```
                                        0x1a,                   /*
FC_BOGUS_STRUCT */
/* 448 */      NdrFcShort( 0x10 ),      /* 16 */
/* 450 */      NdrFcShort( 0x0 ),       /* 0 */
/* 452 */      NdrFcShort( 0x6 ),       /* Offset= 6 (458) */
/* 454 */      0x8,                     /* FC_LONG */
                                        0x39,                   /*
FC_ALIGNM8 */
/* 456 */      0x36,                    /* FC_POINTER */
                                        0x5b,                   /*
FC_END */
/* 458 */

                                        0x11, 0x0,    /* FC_RP */
/* 460 */      NdrFcShort( 0xffffffdc ),/* Offset= -36 (424) */
/* 462 */

                                        0x21,                   /*
FC_BOGUS_ARRAY */

                                        0x3,                    /* 3 */
/* 464 */      NdrFcShort( 0x0 ),       /* 0 */
/* 466 */      0x19,                    /* Corr desc:  field pointer,
FC_ULONG */

                                        0x0,                    /*  */
/* 468 */      NdrFcShort( 0x0 ),       /* 0 */
/* 470 */      NdrFcShort( 0x1 ),       /* Corr flags:  early, */
/* 472 */      NdrFcLong( 0xffffffff ), /* -1 */
/* 476 */      NdrFcShort( 0x0 ),       /* Corr flags:  */
/* 478 */      0x4c,                    /* FC_EMBEDDED_COMPLEX */
                                        0x0,                    /* 0 */
/* 480 */      NdrFcShort( 0xffffff58 ),/* Offset= -168 (312) */
/* 482 */      0x5c,                    /* FC_PAD */
                                        0x5b,                   /*
FC_END */
/* 484 */

                                        0x1a,                   /*
FC_BOGUS_STRUCT */
                                        0x3,                    /* 3 */
/* 486 */      NdrFcShort( 0x10 ),      /* 16 */
/* 488 */      NdrFcShort( 0x0 ),       /* 0 */
/* 490 */      NdrFcShort( 0x6 ),       /* Offset= 6 (496) */
/* 492 */      0x8,                     /* FC_LONG */
                                        0x39,                   /*
FC_ALIGNM8 */
/* 494 */      0x36,                    /* FC_POINTER */
                                        0x5b,                   /*
FC_END */
/* 496 */

                                        0x11, 0x0,    /* FC_RP */
/* 498 */      NdrFcShort( 0xffffffdc ),/* Offset= -36 (462) */
/* 500 */

                                        0x21,                   /*
FC_BOGUS_ARRAY */

                                        0x3,                    /* 3 */
/* 502 */      NdrFcShort( 0x0 ),       /* 0 */
```

```
/* 504 */      0x19,                    /* Corr desc:  field pointer,
FC_ULONG */                             0x0,                    /*  */
/* 506 */      NdrFcShort( 0x0 ),       /* 0 */
/* 508 */      NdrFcShort( 0x1 ),       /* Corr flags:  early, */
/* 510 */      NdrFcLong( 0xffffffff ), /* -1 */
/* 514 */      NdrFcShort( 0x0 ),       /* Corr flags:  */
/* 516 */      0x4c,                    /* FC_EMBEDDED_COMPLEX */
                                        0x0,                    /* 0 */
/* 518 */      NdrFcShort( 0xffffff44 ),/* Offset= -188 (330) */
/* 520 */      0x5c,                    /* FC_PAD */
                                        0x5b,                   /*
FC_END */
/* 522 */

                                        0x1a,                   /*
FC_BOGUS_STRUCT */
                                        0x3,                    /* 3 */
/* 524 */      NdrFcShort( 0x10 ),      /* 16 */
/* 526 */      NdrFcShort( 0x0 ),       /* 0 */
/* 528 */      NdrFcShort( 0x6 ),       /* Offset= 6 (534) */
/* 530 */      0x8,                     /* FC_LONG */
                                        0x39,                   /*
FC_ALIGNM8 */
/* 532 */      0x36,                    /* FC_POINTER */
                                        0x5b,                   /*
FC_END */
/* 534 */

                                        0x11, 0x0,    /* FC_RP */
/* 536 */      NdrFcShort( 0xffffffdc ),/* Offset= -36 (500) */
/* 538 */

                                        0x21,                   /*
FC_BOGUS_ARRAY */

                                        0x3,                    /* 3 */
/* 540 */      NdrFcShort( 0x0 ),       /* 0 */
/* 542 */      0x19,                    /* Corr desc:  field pointer,
FC_ULONG */

                                        0x0,                    /*  */
/* 544 */      NdrFcShort( 0x0 ),       /* 0 */
/* 546 */      NdrFcShort( 0x1 ),       /* Corr flags:  early, */
/* 548 */      NdrFcLong( 0xffffffff ), /* -1 */
/* 552 */      NdrFcShort( 0x0 ),       /* Corr flags:  */
/* 554 */

                                        0x12, 0x0,    /* FC_UP */
/* 556 */      NdrFcShort( 0x176 ),     /* Offset= 374 (930) */
/* 558 */      0x5c,                    /* FC_PAD */
                                        0x5b,                   /*
FC_END */
/* 560 */

                                        0x1a,                   /*
FC_BOGUS_STRUCT */
                                        0x3,                    /* 3 */
/* 562 */      NdrFcShort( 0x10 ),      /* 16 */
/* 564 */      NdrFcShort( 0x0 ),       /* 0 */
/* 566 */      NdrFcShort( 0x6 ),       /* Offset= 6 (572) */
```

```
/* 568 */      0x8,                    /* FC_LONG */
               0x39,                   /*
FC_ALIGNM8 */
/* 570 */      0x36,                   /* FC_POINTER */
               0x5b,                   /*
FC_END */
/* 572 */

               0x11, 0x0,    /* FC_RP */
/* 574 */      NdrFcShort( 0xffffffdc ),  /* Offset= -36 (538) */
/* 576 */

               0x2f,                   /* FC_IP
*/
               0x5a,                   /*
FC_CONSTANT_IID */
/* 578 */      NdrFcLong( 0x2f ),      /* 47 */
/* 582 */      NdrFcShort( 0x0 ),      /* 0 */
/* 584 */      NdrFcShort( 0x0 ),      /* 0 */
/* 586 */      0xc0,                   /* 192 */
               0x0,                    /* 0 */
/* 588 */      0x0,                    /* 0 */
               0x0,                    /* 0 */
/* 590 */      0x0,                    /* 0 */
               0x0,                    /* 0 */
/* 592 */      0x0,                    /* 0 */
               0x46,                   /* 70 */
/* 594 */

               0x1b,                   /*
FC_CARRAY */
               0x0,                    /* 0 */
/* 596 */      NdrFcShort( 0x1 ),      /* 1 */
/* 598 */      0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
               0x0,                    /*  */
/* 600 */      NdrFcShort( 0x4 ),      /* 4 */
/* 602 */      NdrFcShort( 0x1 ),      /* Corr flags:  early, */
/* 604 */      0x1,                    /* FC_BYTE */
               0x5b,                   /*
FC_END */
/* 606 */

               0x1a,                   /*
FC_BOGUS_STRUCT */
               0x3,                    /* 3 */
/* 608 */      NdrFcShort( 0x18 ),     /* 24 */
/* 610 */      NdrFcShort( 0x0 ),      /* 0 */
/* 612 */      NdrFcShort( 0xc ),      /* Offset= 12 (624) */
/* 614 */      0x8,                    /* FC_LONG */
               0x8,                    /*
FC_LONG */
/* 616 */      0x4c,                   /* FC_EMBEDDED_COMPLEX */
               0x0,                    /* 0 */
/* 618 */      NdrFcShort( 0xffffffd6 ),  /* Offset= -42 (576) */
/* 620 */      0x39,                   /* FC_ALIGNM8 */
/* 622 */      0x36,                   /*
/* 624 */      0x5c,                   /* FC_POINTER */
               0x5b,                   /*
FC_END */
/* 626 */      NdrFcShort( 0xfffffffe0 ),  /* Offset= -32 (594) */
/* 628 */

               0x21,                   /*
FC_BOGUS_ARRAY */
               0x3,                    /* 3 */
/* 630 */      NdrFcShort( 0x0 ),      /* 0 */
/* 632 */      0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
               0x0,                    /*  */
/* 634 */      NdrFcShort( 0x0 ),      /* 0 */
/* 636 */      NdrFcShort( 0x1 ),      /* Corr flags:  early, */
/* 638 */      NdrFcLong( 0xffffffff ),  /* -1 */
/* 642 */      NdrFcShort( 0x0 ),      /* Corr flags:  */
/* 644 */

               0x12, 0x0,    /* FC_UP */
/* 646 */      NdrFcShort( 0xffffffd8 ),  /* Offset= -40 (606) */
/* 648 */      0x5c,                   /* FC_PAD */
               0x5b,                   /*
FC_END */
/* 650 */

               0x1a,                   /*
FC_BOGUS_STRUCT */
               0x3,                    /* 3 */
/* 652 */      NdrFcShort( 0x10 ),     /* 16 */
/* 654 */      NdrFcShort( 0x0 ),      /* 0 */
/* 656 */      NdrFcShort( 0x6 ),      /* Offset= 6 (662) */
/* 658 */      0x8,                    /* FC_LONG */
               0x39,                   /*
FC_ALIGNM8 */
/* 660 */      0x36,                   /* FC_POINTER */
               0x5b,                   /*
FC_END */
/* 662 */

               0x11, 0x0,    /* FC_RP */
/* 664 */      NdrFcShort( 0xffffffdc ),  /* Offset= -36 (628) */
/* 666 */

               0x1d,                   /*
FC_SMFARRAY */
               0x0,                    /* 0 */
/* 668 */      NdrFcShort( 0x8 ),      /* 8 */
/* 670 */      0x2,                    /* FC_CHAR */
               0x5b,                   /*
FC_END */
/* 672 */

               0x15,                   /*
FC_STRUCT */
               0x3,                    /* 3 */
/* 674 */      NdrFcShort( 0x10 ),     /* 16 */
               0x6,                    /*
FC_SHORT */
/* 678 */      0x6,                    /* FC_SHORT */
               0x4c,                   /*
FC_EMBEDDED_COMPLEX */
/* 680 */      0x0,                    /* 0 */
               NdrFcShort( 0xfffffff1 ),  /*
Offset= -15 (666) */
               0x5b,                   /*
FC_END */
/* 684 */

               0x1a,                   /*
FC_BOGUS_STRUCT */
               0x3,                    /* 3 */
/* 686 */      NdrFcShort( 0x20 ),     /* 32 */
/* 688 */      NdrFcShort( 0x0 ),      /* 0 */
/* 690 */      NdrFcShort( 0xa ),      /* Offset= 10 (700) */
/* 692 */      0x8,                    /* FC_LONG */
               0x39,                   /*
FC_ALIGNM8 */
/* 694 */      0x36,                   /* FC_POINTER */
               0x4c,                   /*
FC_EMBEDDED_COMPLEX */
/* 696 */      0x0,                    /* 0 */
               NdrFcShort( 0xffffffe7 ),  /*
Offset= -25 (672) */
               0x5b,                   /*
FC_END */
/* 700 */

               0x11, 0x0,    /* FC_RP */
/* 702 */      NdrFcShort( 0xffffff10 ),  /* Offset= -240 (462) */
/* 704 */

               0x1b,                   /*
FC_CARRAY */
               0x0,                    /* 0 */
/* 706 */      NdrFcShort( 0x1 ),      /* 1 */
/* 708 */      0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
               0x0,                    /*  */
/* 710 */      NdrFcShort( 0x0 ),      /* 0 */
/* 712 */      NdrFcShort( 0x1 ),      /* Corr flags:  early, */
/* 714 */      0x1,                    /* FC_BYTE */
               0x5b,                   /*
FC_END */
/* 716 */

               0x1a,                   /*
FC_BOGUS_STRUCT */
               0x3,                    /* 3 */
/* 718 */      NdrFcShort( 0x10 ),     /* 16 */
/* 720 */      NdrFcShort( 0x0 ),      /* 0 */
/* 722 */      NdrFcShort( 0x6 ),      /* Offset= 6 (728) */
/* 724 */      0x8,                    /* FC_LONG */
```

```
                                    0x39,                   /*
FC_ALIGNM8 */
/* 726 */       0x36,

                                    /* FC_POINTER */
                                    0x5b,                   /*
FC_END */
/* 728 */

                                    0x12, 0x0,      /* FC_UP */
/* 730 */       NdrFcShort( 0xffffffe6 ),   /* Offset= -26 (704) */
/* 732 */

                                    0x1b,                   /*
FC_CARRAY */

                                    0x1,                    /* 1 */
/* 734 */       NdrFcShort( 0x2 ),      /* 2 */
/* 736 */       0x19,                   /* Corr desc:  field pointer,
FC_ULONG */

                                    0x0,                    /*  */
/* 738 */       NdrFcShort( 0x0 ),      /* 0 */
/* 740 */       NdrFcShort( 0x1 ),      /* Corr flags:  early, */
/* 742 */       0x6,                    /* FC_SHORT */
                                    0x5b,                   /*

FC_END */
/* 744 */

                                    0x1a,                   /*
FC_BOGUS_STRUCT */

                                    0x3,                    /* 3 */
/* 746 */       NdrFcShort( 0x10 ),     /* 16 */
/* 748 */       NdrFcShort( 0x0 ),      /* 0 */
/* 750 */       NdrFcShort( 0x6 ),      /* Offset= 6 (756) */
/* 752 */       0x8,                    /* FC_LONG */
                                    0x39,                   /*

FC_ALIGNM8 */
/* 754 */       0x36,                   /* FC_POINTER */
                                    0x5b,                   /*

FC_END */
/* 756 */

                                    0x12, 0x0,      /* FC_UP */
/* 758 */       NdrFcShort( 0xffffffe6 ),   /* Offset= -26 (732) */
/* 760 */

                                    0x1b,                   /*
FC_CARRAY */

                                    0x3,                    /* 3 */
/* 762 */       NdrFcShort( 0x4 ),      /* 4 */
/* 764 */       0x19,                   /* Corr desc:  field pointer,
FC_ULONG */

                                    0x0,                    /*  */
/* 766 */       NdrFcShort( 0x0 ),      /* 0 */
/* 768 */       NdrFcShort( 0x1 ),      /* Corr flags:  early, */
/* 770 */       0x8,                    /* FC_LONG */
                                    0x5b,                   /*

FC_END */
/* 772 */
```

```
                                    0x1a,                   /*
FC_BOGUS_STRUCT */                  0x3,                    /* 3 */
/* 774 */       NdrFcShort( 0x10 ),     /* 16 */
/* 776 */       NdrFcShort( 0x0 ),      /* 0 */
/* 778 */       NdrFcShort( 0x6 ),      /* Offset= 6 (784) */
/* 780 */       0x8,                    /* FC_LONG */
                                    0x39,                   /*

FC_ALIGNM8 */                       /* FC_POINTER */
/* 782 */       0x36,                   0x5b,                   /*

FC_END */
/* 784 */

                                    0x12, 0x0,      /* FC_UP */
/* 786 */       NdrFcShort( 0xffffffe6 ),   /* Offset= -26 (760) */
/* 788 */

                                    0x1b,                   /*
FC_CARRAY */

                                    0x7,                    /* 7 */
/* 790 */       NdrFcShort( 0x8 ),      /* 8 */
/* 792 */       0x19,                   /* Corr desc:  field pointer,
FC_ULONG */

                                    0x0,                    /*  */
/* 794 */       NdrFcShort( 0x0 ),      /* 0 */
/* 796 */       NdrFcShort( 0x1 ),      /* Corr flags:  early, */
/* 798 */       0xb,                    /* FC_HYPER */
                                    0x5b,                   /*

FC_END */
/* 800 */

                                    0x1a,                   /*
FC_BOGUS_STRUCT */

                                    0x3,                    /* 3 */
/* 802 */       NdrFcShort( 0x10 ),     /* 16 */
/* 804 */       NdrFcShort( 0x0 ),      /* 0 */
/* 806 */       NdrFcShort( 0x6 ),      /* Offset= 6 (812) */
/* 808 */       0x8,                    /* FC_LONG */
                                    0x39,                   /*

FC_ALIGNM8 */                       /* FC_POINTER */
/* 810 */       0x36,                   0x5b,                   /*

FC_END */
/* 812 */

                                    0x12, 0x0,      /* FC_UP */
/* 814 */       NdrFcShort( 0xffffffe6 ),   /* Offset= -26 (788) */
/* 816 */

                                    0x15,                   /*
FC_STRUCT */
                                    0x3,                    /* 3 */
/* 818 */       NdrFcShort( 0x8 ),      /* 8 */
/* 820 */       0x8,                    /* FC_LONG */
                                    0x8,                    /*

FC_LONG */                          /* FC_PAD */
/* 822 */       0x5c,
```

```
                                    0x5b,                   /*
/* 824 */ FC_END */

                                    0x1b,                   /*
FC_CARRAY */
                                    0x3,                    /* 3 */
/* 826 */       NdrFcShort( 0x8 ),      /* 8 */
/* 828 */       0x7,                    /* Corr desc: FC_USHORT */
                                    0x0,                    /*  */
/* 830 */       NdrFcShort( 0xffc8 ),   /* -56 */
/* 832 */       NdrFcShort( 0x1 ),      /* Corr flags:  early, */
/* 834 */       0x4c,                   /* FC_EMBEDDED_COMPLEX */
                                    0x0,                    /* 0 */
/* 836 */       NdrFcShort( 0xffffffec ),   /* Offset= -20 (816) */
/* 838 */       0x5c,                   /* FC_PAD */
                                    0x5b,                   /*

FC_END */
/* 840 */

                                    0x1a,                   /*
FC_BOGUS_STRUCT */
                                    0x3,                    /* 3 */
/* 842 */       NdrFcShort( 0x38 ),     /* 56 */
/* 844 */       NdrFcShort( 0xffffffec ),   /* Offset= -20 (824) */
/* 846 */       NdrFcShort( 0x0 ),      /* Offset= 0 (846) */
/* 848 */       0x6,                    /* FC_SHORT */
                                    0x6,                    /*

FC_SHORT */
/* 850 */       0x38,                   /* FC_ALIGNM4 */
                                    0x8,                    /*

FC_LONG */
/* 852 */       0x8,                    /* FC_LONG */
                                    0x4c,                   /*

FC_EMBEDDED_COMPLEX */
/* 854 */       0x4,                    /* 4 */
                                    NdrFcShort( 0xfffffe0d ),   /*
Offset= -499 (356) */
                                    0x5b,                   /*

FC_END */
/* 858 */

                                    0x12, 0x0,      /* FC_UP */
/* 860 */       NdrFcShort( 0xffffff02 ),   /* Offset= -254 (606) */
/* 862 */

                                    0x12, 0x8,      /* FC_UP
[simple_pointer] */
/* 864 */       0x1,                    /* FC_BYTE */
                                    0x5c,                   /*

FC_PAD */
/* 866 */

                                    0x12, 0x8,      /* FC_UP
[simple_pointer] */
/* 868 */       0x6,                    /* FC_SHORT */
                                    0x5c,                   /*

FC_PAD */
/* 870 */
```

```
                                0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 872 */      0x8,             /* FC_LONG */
                                0x5c,                   /*
FC_PAD */
/* 874 */
                                0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 876 */      0xa,             /* FC_FLOAT */
                                0x5c,                   /*
FC_PAD */
/* 878 */
                                0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 880 */      0xc,             /* FC_DOUBLE */
                                0x5c,                   /*
FC_PAD */
/* 882 */
                                0x12, 0x0,    /* FC_UP */
/* 884 */      NdrFcShort( 0xfffffda4 ),  /* Offset= -604 (280) */
/* 886 */
                                0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 888 */      NdrFcShort( 0xfffffda6 ),  /* Offset= -602 (286) */
/* 890 */
                                0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 892 */      NdrFcShort( 0xfffffdbc ),  /* Offset= -580 (312) */
/* 894 */
                                0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 896 */      NdrFcShort( 0xfffffdca ),  /* Offset= -566 (330) */
/* 898 */
                                0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 900 */      NdrFcShort( 0xfffffdd8 ),  /* Offset= -552 (348) */
/* 902 */
                                0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 904 */      NdrFcShort( 0x2 ),         /* Offset= 2 (906) */
/* 906 */
                                0x12, 0x0,    /* FC_UP */
/* 908 */      NdrFcShort( 0x16 ),        /* Offset= 22 (930) */
/* 910 */
                                0x15,                   /*
FC_STRUCT */
                                0x7,                    /* 7 */
/* 912 */      NdrFcShort( 0x10 ),        /* 16 */
/* 914 */      0x6,             /* FC_SHORT */
                                0x1,                    /*
FC_BYTE */
/* 916 */      0x1,             /* FC_BYTE */
/* 918 */      0x8,             /* FC_ALIGNM4 */
FC_ALIGNM8 */
/* 920 */      0xb,

FC_END */
/* 922 */

/* 924 */      NdrFcShort( 0xfffffff2 ),
/* 926 */

[simple_pointer] */
/* 928 */      0x2,

FC_PAD */
/* 930 */

FC_BOGUS_STRUCT */

/* 932 */      NdrFcShort( 0x20 ),
/* 934 */      NdrFcShort( 0x0 ),
/* 936 */      NdrFcShort( 0x0 ),
/* 938 */      0x8,

FC_LONG */
/* 940 */      0x6,

FC_SHORT */
/* 942 */      0x6,

FC_SHORT */
/* 944 */      0x4c,

/* 946 */      NdrFcShort( 0xfffffc54 ),
/* 948 */      0x5c,

FC_END */
/* 950 */      0xb4,

/* 952 */      NdrFcShort( 0x0 ),
/* 954 */      NdrFcShort( 0x18 ),
/* 956 */      NdrFcShort( 0x0 ),
/* 958 */      NdrFcShort( 0xfffffc44 ),
/* 960 */

[alloced_on_stack] */
/* 962 */      NdrFcShort( 0x6 ),
/* 964 */

/* 966 */      NdrFcShort( 0xfffffffdc ),
/* 968 */      0xb4,
                                0x38,                   /*
/* FC_LONG */
                                0x39,                   /*

/* FC_HYPER */
                                0x5b,                   /*

                                0x12, 0x0,    /* FC_UP */
/* Offset= -14 (910) */

                                0x12, 0x8,    /* FC_UP

/* FC_CHAR */
                                0x5c,                   /*

                                0x1a,                   /*

                                0x7,                    /* 7 */
/* 32 */
/* 0 */
/* Offset= 0 (936) */
/* FC_LONG */
                                0x8,                    /*

/* FC_SHORT */
                                0x6,                    /*

/* FC_SHORT */
                                0x6,                    /*

/* FC_EMBEDDED_COMPLEX */
                                0x0,                    /* 0 */
/* Offset= -940 (6) */
/* FC_PAD */
                                0x5b,                   /*

/* FC_USER_MARSHAL */
                                0x83,                   /* 131 */
/* 0 */
/* 24 */
/* 0 */
/* Offset= -956 (2) */

                                0x11, 0x4,    /* FC_RP

/* Offset= 6 (968) */

                                0x13, 0x0,    /* FC_OP */
/* Offset= -36 (930) */
/* FC_USER_MARSHAL */
                                0x83,                   /* 131 */
/* 970 */      NdrFcShort( 0x18 ),        /* 24 */
/* 972 */      NdrFcShort( 0x0 ),         /* 0 */
/* 974 */      NdrFcShort( 0x0 ),         /* 0 */
/* 976 */      NdrFcShort( 0xfffffff4 ),  /* Offset= -12 (964) */

                                0x0
        }
    };

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

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

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


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

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{

    if(!_tpcc_com_ps_CHECK_IID(0))
        {
        *pIndex = 0;
        return 1;
        }

    return 0;
}

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

```
    0, /* Filler2 */
    0  /* Filler3 */
};


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

tpcc_com_ps\src\dlldata.c

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

        DO NOT ALTER THIS FILE

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

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

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


#include <rpcproxy.h>

#ifdef __cplusplus
extern "C"   {
#endif

EXTERN_PROXY_FILE( tpcc_com_ps )


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


DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )

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

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

Appendix B Database Load

SETUP.CMD

```
::-------------------------------------------------------------------------------------------
::---  FILE:        RUNSQLCFG.CMD
::---              Microsoft TPC-C Kit Ver. 4.41
::---              Copyright Microsoft, 2001
::---              All Rights Reserved
::---
::---  PURPOSE:    Calls RunSQLCfg.sql to configure SQL Server
::---
::---  ARGUMENTS:  Optionally, the user can pass the following positional
arguments:
::---                Server Name
::---                sa SQL Server account password
::---                Number of Warehouses
::---                Build Option
::---
{full,builddb,objects,objectsfull,bulkload,bulkloadfull,backup}
::---                Database Type
::---                  {normal or scale_down}
::---
::---              If they are not passed, then the user will be prompted by the
VBS file.
::---
::-------------------------------------------------------------------------------------------
@cscript SetupScripts\setup.vbs //H:CScript //I %1 %2 %3 %4 %5
```

SETUP.VBS

```
'-------------------------------------------------------------------------------------------
'---  FILE:        SETUP.VBS
'---              Microsoft TPC-C Kit Ver. 4.41
'---              Copyright Microsoft, 2001
'---              All Rights Reserved
'---
'---  PURPOSE:    This module performs the tasks to create and populate a
TPC-C database
'---
'-------------------------------------------------------------------------------------------
'------------------------------------------
'--- open an windows scripting object
'------------------------------------------
set WshShell = CreateObject("WScript.Shell")
'--------------------------------------------------
'--- before we go any further, make sure that
'--- we are running Windows Scripting Host 5.6
'--- or higher
'--------------------------------------------------
If wScript.Version < 5.6 Then
                wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                wScript.Echo "!!
!!"
                wScript.Echo "!! You do not have the proper version of the
Windows Scripting Host   !!"
                wScript.Echo "!! installed.  Please install the latest Windows
Scripting Host from  !!"
                wScript.Echo "!! ..\tools\wsh\scripten.exe and restart setup.
!!"
                wScript.Echo "!!
!!"
                wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                wScript.Quit
End If
'------------------------------------------
'--- display banner message
'------------------------------------------
wScript.Echo
"*****************************************************************
**************"
wScript.Echo "*                                                              *"
wScript.Echo "*  Microsoft TPC-C Benchmark Kit Ver. 4.41 - Setup
*"
wScript.Echo "*                                                              *"
wScript.Echo
"*****************************************************************
**************"
'----------------------------------------------------
'--- define function to check for any error messages
'----------------------------------------------------
Function CheckSQLOutput(SQL_Out)
                ErrorFlag = 0
                Set SQL_fso = CreateObject("Scripting.FileSystemObject")
                If SQL_fso.FileExists(SQL_Out) Then
                                Set SQL_Out_File =
SQL_fso.OpenTextFile(SQL_Out,1)
                                Do While SQL_Out_File.AtEndOfStream <> True
                                                SQL_Line = SQL_Out_File.ReadLine
                                                'first check to see if the output
contains a message about the login password
                                                If InStr(SQL_Line, "Login failed")
Then
                                                                'display the messages
and get out of here
                                                                ErrorFlag = 1
                                                                wScript.Echo "The
login for userid 'sa' failed."
                                                                wScript.Echo "Please
restart SETUP with the correct password."
                                                Else
                                                                If InStr(SQL_Line,
"Msg") Then
                                                                                'find out
where the "Msg" indicator is in the line
                                                                                LocMsg =
InStr(SQL_Line, "Msg")
                                                                                'find out
where the comma is after the error code

                LocComma = InStr(SQL_Line, ",")
                                                                                'now
isolate the error code

                ErrorCode = Mid(SQL_Line, (LocMsg + 4), (LocComma -
(LocMsg + 4)))
                                                                                Select
Case ErrorCode

                Case " 170"

                                ErrorFlag = 1

                                wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

                                wScript.Echo "Syntax Error."

                                wScript.Echo "SQL Server Error 170."

                                wScript.Echo "Check CREATEDB.SQL."

                                wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

                Case "1801"

                                ErrorFlag = 1

                                wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
```

```
                    wScript.Echo "Database 'tpcc' already exists."

                    wScript.Echo "SQL Server Error 1801."

                    wScript.Echo "Check CREATEDB.SQL."


                    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

          Case "1802"

                    ErrorFlag = 1

                    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

                    wScript.Echo "CREATE DATABASE failed."

                    wScript.Echo "SQL Server Error 1802."

                    wScript.Echo "Check CREATEDB.SQL."

                    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

          Case "1921"

                    ErrorFlag = 1

                    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

                    wScript.Echo "CREATE INDEX failed."

                    wScript.Echo "SQL Server Error 1921."

                    wScript.Echo "Check " & SQL_Out & "."

                    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

          Case "3013"
                              ErrorFlag = 1
                              wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

                    wScript.Echo "BACKUP DATABASE is terminating
abnormally."

                    wScript.Echo "SQL Server Error 3013."

                    wScript.Echo "Check the SQL Server error log for
more details."
                    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

          Case "3201"
                              ErrorFlag = 1
                              wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

                    wScript.Echo "Cannot open backup device."

                    wScript.Echo "Device error or device off-line."

                    wScript.Echo "SQL Server Error 3201."

                    wScript.Echo "See the SQL Server error log for
more details."

                    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

          Case "5105"
                              ErrorFlag = 1
                              wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

                    wScript.Echo "Device Activation Error."

                    wScript.Echo "SQL Server Error 5105."

                    wScript.Echo "Check CREATEDB.SQL."

                    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

          Case "5170"

                    ErrorFlag = 1

                    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

                    wScript.Echo "Cannot create one or more files
because it already exists."

                    wScript.Echo "SQL Server Error 5170."

                    wScript.Echo "Check CREATEDB.SQL."

                    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
          Case "15010","15012"

                    ErrorFlag = 0

          Case "15069"

                    ErrorFlag = 1

                    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

                    wScript.Echo "One or more users are using the
database."

                    wScript.Echo "The requested operation cannot be
completed."

                    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

          Case Else

                    ErrorFlag = 1

                    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

                    wScript.Echo "An error occurred."

                    wScript.Echo "SQL Server Error Code: " &
ErrorCode & "."

                    wScript.Echo "Check " & SQL_Out & " for more
information."

                    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                                                  End
Select
                                        End If
                              End If
                    Loop
                    SQL_Out_File.Close
          End If
          CheckSQLOutput = ErrorFlag
End Function
'-----------------------------------------
'--- end function
'-----------------------------------------
'-------------------------------------------------------
```

```
'--- define function to prompt for user input if necessary
'-------------------------------------------------------------
Function GetUserInput(UserInput)
	Select Case UserInput
		Case "ServerName"
			'--- pre-fill the prompt with the
machine name
			TempServerName =
WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%")
			'--- prompt the use for the setup
particulars
			TempResponse = InputBox("Enter
your server name","TPC-C Setup",TempServerName)
			Do While TempResponse = ""
				rc = MsgBox ("You
must enter a valid server name.",21)
				If rc = 2 Then

	wScript.Echo ""

	wScript.Echo "TPC-C Setup cancelled by user."

	wScript.Quit
				End If
				TempResponse =
WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%")
				TempResponse =
InputBox("Enter your server name","TPC-C Setup",TempServerName)
			Loop
		Case "saPassword"
			TempResponse = InputBox("Enter
the 'sa' password")
		Case "NumberWarehouses"
			TempResponse = InputBox("Enter
the number of warehouses to build","TPC-C Setup")
			Do While TempResponse = ""
				rc = MsgBox ("You
must enter a value for Number of Warehouses.",21)
				If rc = 2 Then

	wScript.Echo ""

	wScript.Echo "TPC-C Setup cancelled by user."

	wScript.Quit
				End If
				TempResponse =
InputBox("Enter the number of warehouses to build","TPC-C Setup")
			Loop
		Case "BuildOption"
			TempResponse = InputBox("Build
Option" & Chr(13) &
"(full,builddb,objects,objectsfull,bulkload,bulkloadfull,backup)","TPC-C
Setup","full")
			Flag = 0
			Do While Flag = 0
				Select Case
TempResponse
					Case
"full","Full","FULL"

		TempResponse = "full"

		Flag = 1
					Case
"builddb","BuildDB","Builddb","BUILDDB"

		TempResponse = "builddb"

		Flag = 1
					Case
"objects","Objects","OBJECTS"

		TempResponse = "objects"

		Flag = 1
					Case
"objectsfull","ObjectsFull","Objectsfull","OBJECTSFULL"

		TempResponse = "objectsfull"

		Flag = 1
					Case
"bulkload","BulkLoad","Bulkload","BULKLOAD"

		TempResponse = "bulkload"

		Flag = 1
					Case
"bulkloadfull","BulkLoadFull","Bulkloadfull","BULKLOADFULL"

		TempResponse = "bulkloadfull"

		Flag = 1
					Case
"backup","Backup","BACKUP"

		TempResponse = "backup"

		Flag = 1
					Case Else

		rc = MsgBox ("Invalid Database Build Option.",21)

		If rc = 2 Then
			wScript.Echo ""

			wScript.Echo "TPC-C Setup cancelled by user."

			wScript.Quit

		End If

		Flag = 0

		TempResponse = InputBox("Build Option" & Chr(13) &
"(full,builddb,objects,objectsfull,bulkload,bulkloadfull,backup)",,"full")


					End Select
			Loop
		Case "DatabaseType"
			TempResponse =
InputBox("Database Type" & Chr(13) & "(normal or scale_down)","TPC-C
Setup","normal")
			'--- set flag
			Flag = 0
			Do While Flag = 0
				Select Case
TempResponse
					Case
"normal","Normal","NORMAL"

		TempResponse = "0"

		Flag = 1
					Case
"scale_down","Scale_Down","Scale_down","SCALE_DOWN"

		TempResponse = "1"

		Flag = 1
					Case Else

		rc = MsgBox ("Invalid Database Type.",21)

		If rc = 2 Then

			wScript.Echo ""

			wScript.Echo "TPC-C Setup cancelled by user."

			wScript.Quit

		End If
```

```
		Flag = 0

		TempResponse = InputBox("Database Type" & Chr(13) &
"(normal or scale_down)",,"normal")
                                                End Select
                                        Loop
		End Select
		GetUserInput = TempResponse
End Function
'-----------------------------------------
'--- end function
'-----------------------------------------
'-------------------------------------------------------
'--- Initialize an array of the TPC-C table names
'-------------------------------------------------------
Dim TableArray(8)
TableArray(0) = "warehouse"
TableArray(1) = "district"
TableArray(2) = "customer"
TableArray(3) = "history"
TableArray(4) = "new_order"
TableArray(5) = "orders"
TableArray(6) = "order_line"
TableArray(7) = "item"
TableArray(8) = "stock"
'-------------------------------------------------------
'--- Initialize an array of the TPC-C build log file names
'-------------------------------------------------------
Dim LogFileArray(21)
LogFileArray(0)  = "version.log"
LogFileArray(1)  = "removedb.log"
LogFileArray(2)  = "createdb.log"
LogFileArray(3)  = "tables.log"
LogFileArray(4)  = "dbopt1.log"
LogFileArray(5)  = "idxordcl.log"
LogFileArray(6)  = "idxitmcl.log"
LogFileArray(7)  = "idxwarcl.log"
LogFileArray(8)  = "idxcuscl.log"
LogFileArray(9)  = "idxnodcl.log"
LogFileArray(10) = "idxdiscl.log"
LogFileArray(11) = "idxstkcl.log"
LogFileArray(12) = "idxodlcl.log"
LogFileArray(13) = "idxcusnc.log"
LogFileArray(14) = "idxhiscl.log"
LogFileArray(15) = "idxordnc.log"
LogFileArray(16) = "bulkload.log"
LogFileArray(17) = "dbopt2.log"
LogFileArray(18) = "nurand_load.log"
LogFileArray(19) = "backupdev.log"
LogFileArray(20) = "backupdev.log"
LogFileArray(21) = "verifyload.log"

'--- open a file system object ------
'------------------------------------------
Set fs = CreateObject("Scripting.FileSystemObject")
'------------------------------------------
'--- grab the current directory value
'------------------------------------------
SetupDirectory = WshShell.CurrentDirectory & "\"
'SetupDirectory = "C:\MSTPCC.441\"
'------------------------------------------
'--- now calculate the other directories
'------------------------------------------
ACIDDirectory = LEFT(SetupDirectory,(LEN(SetupDirectory)-6))
ScriptDirectory = SetupDirectory & "SCRIPTS\"
LogDirectory = SetupDirectory & "LOGS\"
'-------------------------------------------------------------------
'--- now determine if the user passed us any parameters.
'--- the order should be ServerName, sa Password, Number of Warehouses,
'--- Build Option, and Database Type
'-------------------------------------------------------------------
Set objArgs = wScript.Arguments
Select Case objArgs.Length
	Case 0
		'------------------------------------------
		'--- get the server name
		'------------------------------------------
		ServerName = GetUserInput("ServerName")
		'------------------------------------------
		'--- get the sa password
		'------------------------------------------
		saPassword = GetUserInput("saPassword")
		'------------------------------------------
		'--- get the number of warehouses
		'------------------------------------------
		NumberWarehouses =
GetUserInput("NumberWarehouses")
		'------------------------------------------
		'--- get the build option
		'------------------------------------------
		BuildOption = GetUserInput("BuildOption")
		'------------------------------------------
		'--- get the database type
		'------------------------------------------
		DatabaseType = GetUserInput("DatabaseType")
	Case 1
		'-------------------------------------------------------
		'--- assume that the server name was passed
correctly
		'-------------------------------------------------------
		'------------------------------------------
		'--- store the server name
		'------------------------------------------
		ServerName = objArgs(0)
		'------------------------------------------
		'--- get the sa password
		'------------------------------------------
		saPassword = GetUserInput("saPassword")
		'------------------------------------------
		'--- get the number of warehouses
		'------------------------------------------
		NumberWarehouses =
GetUserInput("NumberWarehouses")
		'------------------------------------------
		'--- get the build option
		'------------------------------------------
		BuildOption = GetUserInput("BuildOption")
		'------------------------------------------
		'--- get the database type
		'------------------------------------------
		DatabaseType = GetUserInput("DatabaseType")
		If DatabaseType = "scale_down" or DatabaseType
= "Scale_Down" or DatabaseType = "Scale_down" Then
			DatabaseType = 1
		Else
			DatabaseType = 0
		End If
	Case 2
		'-------------------------------------------------------------------
-------
		'--- assume that the server name and sa password
was passed correctly
		'-------------------------------------------------------------------
-------
		'------------------------------------------
		'--- store the server name
		'------------------------------------------
		ServerName = objArgs(0)
		'------------------------------------------
		'--- store the sa password
		'------------------------------------------
		saPassword = objArgs(1)
		'------------------------------------------
		'--- get the number of warehouses
		'------------------------------------------
		NumberWarehouses =
GetUserInput("NumberWarehouses")
		'------------------------------------------
		'--- get the build option
		'------------------------------------------
		BuildOption = GetUserInput("BuildOption")
		'------------------------------------------
		'--- get the database type
		'------------------------------------------
		DatabaseType = GetUserInput("DatabaseType")
		If DatabaseType = "scale_down" or DatabaseType
= "Scale_Down" or DatabaseType = "Scale_down" Then
			DatabaseType = 1
		Else
```

```
                        DatabaseType = 0
                End If
        Case 3
                '-------------------------------------------------------------------
---------------------------
                '--- assume that the server name,sa password,
and number of warehouses was passed correctly
                '-------------------------------------------------------------------
---------------------------
                '-----------------------------------------
                '--- store the server name
                '-----------------------------------------
                ServerName = objArgs(0)
                '-----------------------------------------
                '--- store the sa password
                '-----------------------------------------
                saPassword = objArgs(1)
                '-----------------------------------------
                '--- store the number of warehouses
                '-----------------------------------------
                NumberWarehouses = objArgs(2)
                '-----------------------------------------
                '--- get the build option
                '-----------------------------------------
                BuildOption = GetUserInput("BuildOption")
                '-----------------------------------------
                '--- get the database type
                '-----------------------------------------
                DatabaseType = GetUserInput("DatabaseType")
                If DatabaseType = "scale_down" or DatabaseType
= "Scale_Down" or DatabaseType = "Scale_down" Then
                        DatabaseType = 1
                Else
                        DatabaseType = 0
                End If
        Case 4
                '-------------------------------------------------------------------
-----------------------------------------
                '--- assume that the server name,sa
password,number of warehouses, and build option was passed correctly
                '-------------------------------------------------------------------
-----------------------------------------
                '-----------------------------------------
                '--- store the server name
                '-----------------------------------------
                ServerName = objArgs(0)
                '-----------------------------------------
                '--- store the sa password
                '-----------------------------------------
                saPassword = objArgs(1)
                '-----------------------------------------
                '--- store the number of warehouses
                NumberWarehouses = objArgs(2)
                '-----------------------------------------
                '--- store the build option
                '-----------------------------------------
                BuildOption = objArgs(3)
                '-----------------------------------------
                '--- get the database type
                '-----------------------------------------
                DatabaseType = GetUserInput("DatabaseType")
                If DatabaseType = "scale_down" or DatabaseType
= "Scale_Down" or DatabaseType = "Scale_down" Then
                        DatabaseType = 1
                Else
                        DatabaseType = 0
                End If
        Case 5
                '-------------------------------------------------------
                '--- assume all the parameters were passed in
correctly
                '-------------------------------------------------------
                '-----------------------------------------
                '--- store the server name
                '-----------------------------------------
                ServerName = objArgs(0)
                '-----------------------------------------
                '--- store the sa password
                '-----------------------------------------
                saPassword = objArgs(1)
                '-----------------------------------------
                '--- store the number of warehouses
                '-----------------------------------------
                NumberWarehouses = objArgs(2)
                '-----------------------------------------
                '--- store the build option
                '-----------------------------------------
                BuildOption = objArgs(3)
                '-----------------------------------------
                '--- get the database type
                '-----------------------------------------
                DatabaseType = objArgs(4)
                If DatabaseType = "scale_down" or DatabaseType
= "Scale_Down" or DatabaseType = "Scale_down" Then
                        DatabaseType = 1
                Else
                        DatabaseType = 0
                End If
End Select
'-------------------------------------------------------------
'--- now that we have all the variables filled in, let's get to work
'--- cleanup any old .err files
'-------------------------------------------------------------
For i = 0 to 8
        If fs.FileExists(LogPath & TableArray(i) & ".err") Then
                fs.DeleteFile LogPath & TableArray(i) & ".err"
        End If
Next
For i = 0 to 21
        If fs.FileExists(LogPath & LogFileArray(i)) Then
                fs.DeleteFile LogPath & LogFileArray(i)
        End If
Next
'-------------------------------------------------------------
'--- now grab the version of SQL Server you are running this against
'-------------------------------------------------------------
Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S" &
ServerName & " -e -i" & ScriptDirectory & "utility\version.sql -o" &
LogDirectory & "version.log")
Do While oExec.Status = 0
    wScript.Sleep 100
Loop
rc = CheckSQLOutput(LogDirectory & "version.log")
If rc <> 0 Then
        wScript.Quit
End If
If (BuildOption = "full" OR BuildOption = "builddb") Then
        wScript.Echo "Removing any existing TPCC database and
backup devices..."
        Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ScriptDirectory & NumberWarehouses &
".war\database\removedb.sql -o" & LogDirectory & "removedb.log")
        Do While oExec.Status = 0
            wScript.Sleep 100
        Loop
        rc = CheckSQLOutput(LogDirectory & "removedb.log")
        If rc <> 0 Then
                wScript.Quit
        End If
        wScript.Echo "Building database files and database..."
        Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ScriptDirectory & NumberWarehouses &
".war\database\createdb.sql -o" & LogDirectory & "createdb.log")
        Do While oExec.Status = 0
            wScript.Sleep 100
        Loop
        rc = CheckSQLOutput(LogDirectory & "createdb.log")
        If rc <> 0 Then
                wScript.Quit
        End If
End If
'-----------------------------------------
'--- build tables and stored procedures
'-----------------------------------------
If (BuildOption = "full" OR BuildOption = "builddb" _
    OR BuildOption = "objects" OR BuildOption = "objectsfull") Then
        wScript.Echo "Creating TPC-C database tables..."
```

```
		Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ScriptDirectory & NumberWarehouses &
".war\ddl\tables.sql -o" & LogDirectory & "tables.log")
		Do While oExec.Status = 0
		    wScript.Sleep 100
		Loop
		rc = CheckSQLOutput(LogDirectory & "tables.log")
		If rc <> 0 Then
			wScript.Quit
		End If
		wScript.Echo "Creating database objects..."
		Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ScriptDirectory & "dml\neword.sql -o" &
LogDirectory & "sp_neword.log")
		Do While oExec.Status = 0
		    wScript.Sleep 100
		Loop
		rc = CheckSQLOutput(LogDirectory & "sp_neword.log")
		If rc <> 0 Then
			wScript.Quit
		End If
		Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ScriptDirectory & "dml\payment.sql -o" &
LogDirectory & "sp_payment.log")
		Do While oExec.Status = 0
		    wScript.Sleep 100
		Loop
		rc = CheckSQLOutput(LogDirectory & "sp_payment.log")
		If rc <> 0 Then
			wScript.Quit
		End If
		Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ScriptDirectory & "dml\ordstat.sql -o" &
LogDirectory & "sp_ordstat.log")
		Do While oExec.Status = 0
		    wScript.Sleep 100
		Loop
		rc = CheckSQLOutput(LogDirectory & "sp_ordstat.log")
		If rc <> 0 Then
			wScript.Quit
		End If
		Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ScriptDirectory & "dml\delivery.sql -o" &
LogDirectory & "sp_delivery.log")
		Do While oExec.Status = 0
		    wScript.Sleep 100
		Loop
		rc = CheckSQLOutput(LogDirectory & "sp_delivery.log")
		If rc <> 0 Then
			wScript.Quit
		End If
```

```
		Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ScriptDirectory & "dml\stocklev.sql -o" &
LogDirectory & "sp_stocklev.log")
		Do While oExec.Status = 0
		    wScript.Sleep 100
		Loop
		rc = CheckSQLOutput(LogDirectory & "sp_stocklev.log")
		If rc <> 0 Then
			wScript.Quit
		End If
		Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ScriptDirectory & "dml\version.sql -o" &
LogDirectory & "sp_version.log")
		Do While oExec.Status = 0
		    wScript.Sleep 100
		Loop
		rc = CheckSQLOutput(LogDirectory & "sp_version.log")
		If rc <> 0 Then
			wScript.Quit
		End If
		wScript.Echo "Database object creation complete..."
End If
If (BuildOption = "full" OR BuildOption = "builddb" _
    OR BuildOption = "objects" OR BUildOption = "objectsfull" _
    OR BuildOption = "bulkload" OR BuildOption = "bulkloadfull") Then
		wScript.Echo "Setting database options before load..."
		Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ScriptDirectory & "utility\dbopt1.sql -o" &
LogDirectory & "dbopt1.log")
		Do While oExec.Status = 0
		    wScript.Sleep 100
		Loop
		rc = CheckSQLOutput(LogDirectory & "dbopt1.log")
		If rc <> 0 Then
			wScript.Quit
		End If
		'-------------------------------------------------------
		'--- before we start tpccldr.exe, check the registry
		'--- to ensure that the Shared Memory Protocol is off.
		'--- if it is on, store the setting so we can return
		'--- the system to the pre-tpccldr state.
		'-------------------------------------------------------
		SharedMemoryRegKey =
WshShell.RegRead("HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLSer
ver\Client\SharedMemoryOn")
		If SharedMemoryRegKey = 1 Then
			WshShell.RegWrite
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SharedM
emoryOn", 0, "REG_DWORD"
		End If
		wScript.Echo "Beginning data load and index creation..."
		CMD_String = SetupDirectory & "\loader\bin\tpccldr.exe"
		CMD_String = CMD_String & " -S" & ServerName
		CMD_String = CMD_String & " -Usa"
```

```
		CMD_String = CMD_String & " -P" & saPassword
		CMD_String = CMD_String & " -W" & NumberWarehouses
		CMD_String = CMD_String & " -F" & LogDirectory &
"bulkload.log"
		'CMD_String = CMD_String & " -L" & LogDirectory
		CMD_String = CMD_String & " -d" & ScriptDirectory &
NumberWarehouses & ".war\ddl"
		'CMD_String = CMD_String & " -c" & DatabaseType
		oExec = WshShell.Run(CMD_String, 2, true)
		'-------------------------------------------------------
		'--- now that the loader is finished, put the
		'--- SharedMemoryOn registry key back to its original
		'--- value.
		'-------------------------------------------------------
		If SharedMemoryRegKey = 1 Then
			WshShell.RegWrite
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SharedM
emoryOn", 1, "REG_DWORD"
		End If
		wScript.Echo "Setting database options after load..."
		Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ScriptDirectory & "utility\dbopt2.sql -o" &
LogDirectory & "dbopt2.log")
		Do While oExec.Status = 0
		    wScript.Sleep 100
		Loop
		rc = CheckSQLOutput(LogDirectory & "dbopt2.log")
		If rc <> 0 Then
			wScript.Quit
		End If
		wScript.Echo "Data load and index creation complete."
		'-----------------------------------------
		'--- now parse the index creation logs
		'--- to see if there were any errors
		'--- there.
		'-----------------------------------------
		For i = 5 to 15
			rc = CheckSQLOutput(LogDirectory &
LogFileArray(i))
			If rc <> 0 Then
				wScript.Quit
			End If
		Next
		wScript.Echo "Calculating initial database space usage...."
		Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ACIDDirectory & "space\scripts\spused.sql -o" &
ACIDDirectory & "space\spused.ver")
		Do While oExec.Status = 0
		    wScript.Sleep 100
		Loop
		Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ACIDDirectory & "space\scripts\splog.sql -o" &
ACIDDirectory & "space\splog.ver")
		Do While oExec.Status = 0
```

```
            wScript.Sleep 100
        Loop
        Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ACIDDirectory & "space\scripts\spfiles.sql -o" &
ACIDDirectory & "space\spfiles.ver")
        Do While oExec.Status = 0
            wScript.Sleep 100
        Loop
        '----------------------------------------
        '--- now that the loader is finished
        '--- check the .err files and if they
        '--- are of zero length, delete them.
        '----------------------------------------
        Set fsErr = CreateObject("Scripting.FileSystemObject")
        Set fErr = fsErr.GetFolder(LogDirectory)
        Set fcErr = fErr.Files
        For Each f1 In fcErr
                If f1.Type = "ERR File" Then
                        If f1.Size = 0 Then
                                f1.Delete
                        End If
                End If
        Next
        Set fcErr = Nothing
        Set fErr = Nothing
        Set fsErr = Nothing
End If
If (BuildOption = "full" _
   OR BUildOption = "objectsfull" _
   OR BuildOption = "bulkloadfull" _
   OR BuildOption = "backup") Then
        wScript.Echo "Creating Backup Device(s)..."
        Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ScriptDirectory & NumberWarehouses &
".war\database\backupdev.sql -o" & LogDirectory & "backupdev.log")
        Do While oExec.Status = 0
            wScript.Sleep 100
        Loop
        rc = CheckSQLOutput(LogDirectory & "backupdev.log")
        If rc <> 0 Then
                wScript.Quit
        End If
        wScript.Echo "Backing up database..."
        Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ScriptDirectory & NumberWarehouses &
".war\database\backup.sql -o" & LogDirectory & "backup.log")
        Do While oExec.Status = 0
            wScript.Sleep 100
        Loop
        rc = CheckSQLOutput(LogDirectory & "backup.log")
        If rc <> 0 Then
                wScript.Quit
        wScript.Echo "Database backup complete."
        End If
End If
If (BuildOption = "full" _
   OR BUildOption = "objectsfull" _
   OR BuildOption = "bulkloadfull") Then
        wScript.Echo "Verifying TPC-C database load..."
        Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -S"
& ServerName & " -e -i" & ScriptDirectory & "utility\verifytpccload.sql -o" &
LogDirectory & "verifyload.log")
        Do While oExec.Status = 0
            wScript.Sleep 100
        Loop
        rc = CheckSQLOutput(LogDirectory & "verifyload.log")
        If rc <> 0 Then
                wScript.Quit
        End If
        wScript.Echo "Check logs\verifyload.log to verify database
load."
End If
'----------------------------------------
'--- display banner message
'----------------------------------------
wScript.Echo
"*************************************************************
*************"
wScript.Echo "*                                                    *"
wScript.Echo "*  Microsoft TPC-C Benchmark Kit Ver. 4.41 - Setup Complete
*"
wScript.Echo "*                                                    *"
wScript.Echo
"*************************************************************
*************"
```

CREATEDB.SQL

```
-- File:    CREATEDB.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.41
--          Copyright Microsoft, 2001
-- Purpose:  Creates tpcc database and backup files


use master
go

--              Create temporary table for timing

if exists ( select name from sysobjects where name = 'tpcc_timer' )
        drop table tpcc_timer
go

create table tpcc_timer
(
        start_date                                      char(30),
        end_date                                        char(30)
)

insert  into tpcc_timer values (0,0)
go

--      Store starting time

update  tpcc_timer
set     start_date    = (select convert(char(30), getdate(),9))
go

--  create main database files

CREATE DATABASE tpcc
ON PRIMARY
(       NAME                    = MSSQL_tpcc_root,
        FILENAME    = "C:\MSSQL_tpcc_root.mdf",
        SIZE                    = 8MB,
        FILEGROWTH              = 0),

FILEGROUP  MSSQL_misc_fg
      (NAME=tpcc_misc1, FILENAME="c:\mnt\misc1\",
SIZE=52000MB,FILEGROWTH=0),
        (NAME=tpcc_misc2, FILENAME="c:\mnt\misc2\",
SIZE=52000MB,FILEGROWTH=0),
        (NAME=tpcc_misc3, FILENAME="c:\mnt\misc3\",
SIZE=52000MB,FILEGROWTH=0),
        (NAME=tpcc_misc4, FILENAME="c:\mnt\misc4\",
SIZE=52000MB,FILEGROWTH=0),

FILEGROUP  MSSQL_cs_fg
        (NAME=tpcc_cs1, FILENAME="c:\mnt\cs1\",
SIZE=101500MB,FILEGROWTH=0),
        (NAME=tpcc_cs2, FILENAME="c:\mnt\cs2\",
SIZE=101500MB,FILEGROWTH=0),
        (NAME=tpcc_cs3, FILENAME="c:\mnt\cs3\",
SIZE=101500MB,FILEGROWTH=0),
        (NAME=tpcc_cs4, FILENAME="c:\mnt\cs4\",
SIZE=101500MB,FILEGROWTH=0)

LOG ON
(       NAME                    = MSSQL_tpcc_log,
        FILENAME    = "L:",
        SIZE                    = 75000MB,
        FILEGROWTH              = 0)
```

```
COLLATE Latin1_General_BIN
go

-- Store ending time
update          tpcc_timer
set             end_date      = (select convert(char(30), getdate(),9))
go

select "Elapsed time (in seconds): ", datediff(second,(select start_date from
tpcc_timer),(select end_date from tpcc_timer))

--              remove temporary table

if exists ( select name from sysobjects where name = 'tpcc_timer' )
                drop table tpcc_timer
go
```

BACKUP.SQL

```
-- File:    BACKUP.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.41
--          Copyright Microsoft, 2001
-- Purpose:  Creates backup of tpcc database

backup database tpcc to
disk='w:\tpcc1.bak',
disk='x:\tpcc2.bak',
disk='y:\tpcc3.bak',
disk='z:\tpcc4.bak'
with init, maxtransfersize=1048576, stats = 1
go
```

BACKUPDEV.SQL

```
-- File:    BACKUPDEVB.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.41
--          Copyright Microsoft, 2001
-- Purpose:  Creates tpcc database Backup Devices


use master
go

-- create backup devices

-- no devices needed
```

RESTORE.SQL

```
-- File:    RESTORE.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.41
--          Copyright Microsoft, 2001
-- Purpose:  Loads database backup from backup files


declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

load database tpcc from tpccback1, tpccback2, tpccback3, tpccback4, with
stats = 1

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

IDXCUSCL.SQL

```
-- File:    IDXCUSCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.41
--          Copyright Microsoft, 2001
-- Purpose:  Creates clustered index on customer table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'customer_c1' )
                drop index customer.customer_c1

create unique clustered index customer_c1 on customer(c_w_id, c_d_id,
c_id)
                on MSSQL_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

IDXCUSNC.SQL

```
-- File:    IDXCUSNC.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.41
--          Copyright Microsoft, 2001
-- Purpose:  Creates non-clustered index on customer table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'customer_nc1' )
                drop index customer.customer_nc1

create unique nonclustered index customer_nc1 on customer(c_w_id, c_d_id,
c_last, c_first, c_id)
                on MSSQL_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

IDXDISCL.SQL

```
-- File:    IDXDISCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.41
--          Copyright Microsoft, 2001
-- Purpose:  Creates clustered index on district table


use tpcc
```

```
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'district_c1' )
                drop index district.district_c1

create unique clustered index  district_c1 on district(d_w_id, d_id)
                with fillfactor=100 on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

IDXITMCL.SQL

```
-- File:     IDXITMCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:  Creates clustered index on item table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'item_c1' )
                drop index item.item_c1

create unique clustered index item_c1 on item(i_id)
                on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

IDXNODCL.SQL

```
-- File:     IDXNODCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:  Creates clustered index on new_order table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'new_order_c1' )
                drop index new_order.new_order_c1

create unique clustered index new_order_c1 on new_order(no_w_id,
no_d_id, no_o_id)
                on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

IDXODLCL.SQL

```
-- File:     IDXODLCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:  Creates clustered index on order_line table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'order_line_c1' )
                drop index order_line.order_line_c1

create unique clustered index order_line_c1 on order_line(ol_w_id, ol_d_id,
ol_o_id, ol_number)
                on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

IDXORDCL.SQL

```
-- File:     IDXORDCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:  Creates clustered index on orders table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'orders_c1' )
                drop index orders.orders_c1

create unique clustered index orders_c1 on orders(o_w_id, o_d_id, o_id)
                on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

IDXORDNC.SQL

```
-- File:     IDXORDNC.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:  Creates non-clustered index on orders table
```

```
use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'orders_nc1' )
                drop index orders.orders_nc1

create index orders_nc1 on orders(o_w_id, o_d_id, o_c_id, o_id)
                on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

IDXSTKCL.SQL

```
-- File:     IDXSTKCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:  Creates clustered index on stock table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'stock_c1' )
                drop index stock.stock_c1

create unique clustered index stock_c1 on stock(s_i_id, s_w_id)
                on MSSQL_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

IDXWARCL.SQL

```
-- File:     IDXWARCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:  Creates clustered index on warehouse table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'warehouse_c1' )
                drop index warehouse.warehouse_c1

create unique clustered index warehouse_c1 on warehouse(w_id)
                with fillfactor=100 on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

B.1 Database Options

DBOPT1.SQL

```
-- File:     DBOPT1.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:  Sets database options for data load


use master
go

exec sp_dboption tpcc,'select into/bulkcopy',true
exec sp_dboption tpcc,'trunc. log on chkpt.',true
exec sp_dboption tpcc,'torn page detection',false
go
use tpcc
go

checkpoint
go
```

DBOPT2.SQL

```
-- File:     DBOPT2.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:  Resets database options after data load

exec sp_dboption tpcc,'select into/bulkcopy',false
exec sp_dboption tpcc,'trunc. log on chkpt.',false
exec sp_dboption tpcc,'torn page detection',false
GO

USE tpcc
GO

CHECKPOINT
GO

sp_configure 'allow updates',1
GO

RECONFIGURE WITH OVERRIDE
GO

DECLARE     @msg                        varchar(50)

--                                      --
--       OPTIONS FOR SQL SERVER 2000        --
-- Set option values for user-defined indexes --
--                                      --

SET         @msg        = ' '
PRINT       @msg
SET         @msg        = 'Setting SQL Server indexoptions'
PRINT       @msg
SET         @msg        = ' '
PRINT       @msg

EXEC sp_indexoption     'customer',    'DisAllowPageLocks',     TRUE
EXEC sp_indexoption     'district',    'DisAllowPageLocks',     TRUE
EXEC sp_indexoption     'warehouse',   'DisAllowPageLocks',     TRUE
```

```
EXEC sp_indexoption     'stock',        'DisAllowPageLocks',        TRUE
EXEC sp_indexoption     'order_line',   'DisAllowRowLocks',         TRUE
EXEC sp_indexoption     'orders',       'DisAllowRowLocks',         TRUE
EXEC sp_indexoption     'new_order',    'DisAllowRowLocks',         TRUE
EXEC sp_indexoption     'item',                         'DisAllowRowLocks',
                TRUE
EXEC sp_indexoption     'item',                         'DisAllowPageLocks',
                TRUE
GO

Print ' '
Print '******************'
Print 'Pre-specified Locking Hierarchy:'
Print '             Lockflag = 0 ==> No pre-specified hierarchy'
Print '   Lockflag = 1 ==> Lock at Page-level then Table-level'
Print '   Lockflag = 2 ==> Lock at Row-level then Table-level'
Print '   Lockflag = 3 ==> Lock at Table-level'
Print ' '

SELECT      name,lockflags
FROM        sysindexes
WHERE       object_id('warehouse')      = id OR
            object_id('district')       = id OR
            object_id('customer')       = id OR
            object_id('stock')          = id OR
            object_id('orders')         = id OR
            object_id('order_line')     = id OR
            object_id('history')        = id OR
            object_id('new_order')      = id OR
            object_id('item')           = id
ORDER       BY lockflags asc
GO

sp_configure 'allow updates',0
GO

RECONFIGURE WITH OVERRIDE
GO

EXEC sp_dboption tpcc,    'auto update statistics',     FALSE
EXEC sp_dboption tpcc,    'auto create statistics',     FALSE
GO

EXEC sp_tableoption     'district',     'pintable',true
EXEC sp_tableoption     'warehouse',    'pintable',true
EXEC sp_tableoption     'new_order',    'pintable',true
EXEC sp_tableoption     'item',                         'pintable',true
GO
```

B.2 Table definitions

table.SQL

```
-- File:    TABLES.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Creates TPC-C tables
-- Not Null for 64-bit OSQL

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    NOT NULL,
            w_name                  char(10)    NOT NULL,
            w_street_1  char(20)    NOT NULL,
            w_street_2  char(20)    NOT NULL,
            w_city                  char(20)    NOT NULL,
            w_state                 char(2)                 NOT
NULL,
            w_zip                   char(9)                 NOT
NULL,
            w_tax                   numeric(4,4) NOT NULL,
            w_ytd                   numeric(12,2)           NOT
NULL
) on MSSQL_misc_fg
go

create table district
(
            d_id                    tinyint     NOT NULL,
            d_w_id                  smallint    NOT NULL,
            d_name                  char(10)    NOT NULL,
            d_street_1  char(20)    NOT NULL,
            d_street_2  char(20)    NOT NULL,
            d_city                  char(20)    NOT NULL,
            d_state                 char(2)     NOT NULL,
            d_zip                   char(9)     NOT NULL,
            d_tax                   numeric(4,4) NOT NULL,
            d_ytd                   numeric(12,2)           NOT
NULL,
            d_next_o_id int         NOT NULL
) on MSSQL_misc_fg
go

create table customer
(
            c_id                    int         NOT NULL,
            c_d_id                  tinyint     NOT NULL,
            c_w_id                  smallint    NOT NULL,
            c_first                 char(16)    NOT NULL,
            c_middle                char(2)     NOT NULL,
            c_last                  char(16)    NOT NULL,
            c_street_1  char(20)    NOT NULL,
            c_street_2  char(20)    NOT NULL,
            c_city                  char(20)    NOT NULL,
            c_state                 char(2)     NOT NULL,
            c_zip                   char(9)     NOT NULL,
            c_phone                 char(16)    NOT NULL,
            c_since                 datetime    NOT NULL,
            c_credit                char(2)     NOT NULL,
            c_credit_lim numeric(12,2)          NOT NULL,
            c_discount  numeric(4,4) NOT NULL,
            c_balance   numeric(12,2)           NOT NULL,
            c_ytd_payment           numeric(12,2)           NOT
NULL,
            c_payment_cnt           smallint    NOT NULL,
            c_delivery_cnt          smallint    NOT NULL,
            c_data                  char(500)   NOT NULL
) on MSSQL_cs_fg
```

```
go

create table history
(
        h_c_id          int             NOT NULL,
        h_c_d_id        tinyint         NOT NULL,
        h_c_w_id        smallint        NOT NULL,
        h_d_id          tinyint         NOT NULL,
        h_w_id          smallint        NOT NULL,
        h_date          datetime        NOT NULL,
        h_amount        numeric(6,2) NOT NULL,
        h_data          char(24)        NOT NULL
) on MSSQL_misc_fg
go

create table new_order
(
        no_o_id         int             NOT NULL,
        no_d_id         tinyint         NOT NULL,
        no_w_id         smallint        NOT NULL
) on MSSQL_misc_fg
go

create table orders
(
        o_id            int             NOT NULL,
        o_d_id          tinyint         NOT NULL,
        o_w_id          smallint        NOT NULL,
        o_c_id          int             NOT NULL,
        o_entry_d       datetime        NOT NULL,
        o_carrier_id    tinyint         NOT NULL,
        o_ol_cnt        tinyint         NOT NULL,
        o_all_local     tinyint         NOT NULL
) on MSSQL_misc_fg
go

create table order_line
(
        ol_o_id         int             NOT NULL,
        ol_d_id         tinyint         NOT NULL,
        ol_w_id         smallint        NOT NULL,
        ol_number       tinyint         NOT NULL,
        ol_i_id         int             NOT NULL,
        ol_supply_w_id  smallint        NOT NULL,
        ol_delivery_d   datetime        NOT NULL,
        ol_quantity     smallint        NOT NULL,
        ol_amount       numeric(6,2) NOT NULL,
        ol_dist_info    char(24)        NOT NULL
) on MSSQL_misc_fg
go

create table item
(
        i_id            int             NOT NULL,
        i_im_id         int             NOT NULL,
        i_name          char(24)        NOT NULL,
        i_price         numeric(5,2) NOT NULL,
        i_data          char(50)        NOT NULL
) on MSSQL_misc_fg
go

create table stock
(
        s_i_id          int             NOT NULL,
        s_w_id          smallint        NOT NULL,
        s_quantity      smallint        NOT NULL,
        s_dist_01       char(24)        NOT NULL,
        s_dist_02       char(24)        NOT NULL,
        s_dist_03       char(24)        NOT NULL,
        s_dist_04       char(24)        NOT NULL,
        s_dist_05       char(24)        NOT NULL,
        s_dist_06       char(24)        NOT NULL,
        s_dist_07       char(24)        NOT NULL,
        s_dist_08       char(24)        NOT NULL,
        s_dist_09       char(24)        NOT NULL,
        s_dist_10       char(24)        NOT NULL,
        s_ytd           int             NOT NULL,
        s_order_cnt     smallint        NOT NULL,
        s_remote_cnt    smallint        NOT NULL,
        s_data          char(50)        NOT NULL
) on MSSQL_cs_fg
go
```

B.3 Stored Procedures

NEWORD.SQL

```
-- File:    NEWORD.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.41
--          Copyright Microsoft, 2001
-- Purpose:  Creates new order transaction stored procedure
--
--              Interface Level: 4.10.000

use tpcc
go

if exists ( select name from sysobjects where name = 'tpcc_neworder' )
go              drop procedure tpcc_neworder

create proc tpcc_neworder
                                                        @w_id
smallint,
                                                        @d_id
tinyint,
                                                        @c_id
int,
                @o_ol_cnt       tinyint,

                @o_all_local    tinyint,
                                                        @i_id1
int = 0, @s_w_id1  smallint = 0, @ol_qty1  smallint = 0,
                                                        @i_id2
int = 0, @s_w_id2  smallint = 0, @ol_qty2  smallint = 0,
                                                        @i_id3
int = 0, @s_w_id3  smallint = 0, @ol_qty3  smallint = 0,
                                                        @i_id4
int = 0, @s_w_id4  smallint = 0, @ol_qty4  smallint = 0,
                                                        @i_id5
int = 0, @s_w_id5  smallint = 0, @ol_qty5  smallint = 0,
                                                        @i_id6
int = 0, @s_w_id6  smallint = 0, @ol_qty6  smallint = 0,
                                                        @i_id7
int = 0, @s_w_id7  smallint = 0, @ol_qty7  smallint = 0,
                                                        @i_id8
int = 0, @s_w_id8  smallint = 0, @ol_qty8  smallint = 0,
                                                        @i_id9
int = 0, @s_w_id9  smallint = 0, @ol_qty9  smallint = 0,
                                                        @i_id10
int = 0, @s_w_id10 smallint = 0, @ol_qty10 smallint = 0,
                                                        @i_id11
int = 0, @s_w_id11 smallint = 0, @ol_qty11 smallint = 0,
                                                        @i_id12
int = 0, @s_w_id12 smallint = 0, @ol_qty12 smallint = 0,
                                                        @i_id13
int = 0, @s_w_id13 smallint = 0, @ol_qty13 smallint = 0,
                                                        @i_id14
int = 0, @s_w_id14 smallint = 0, @ol_qty14 smallint = 0,
                                                        @i_id15
int = 0, @s_w_id15 smallint = 0, @ol_qty15 smallint = 0



as
declare         @w_tax          numeric(4,4),
        @d_tax          numeric(4,4),
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     numeric(4,4),
        @i_price        numeric(5,2),
```

```
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
                @li_no          int,
                @o_id                           int,
                @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      smallint,
        @li_qty         smallint,
                @ol_number int,
                @c_id_local  int

begin

begin transaction n

-- get district tax and next availible order id and update
-- plus initialize local variables

                update          district
                set             @d_tax       = d_tax,
                                @o_id        = d_next_o_id,
                                d_next_o_id  = d_next_o_id + 1,
                                @o_entry_d   = getdate(),
                                @li_no       = 0,
                                @commit_flag = 1
                where           d_w_id       = @w_id and
                                d_id         = @d_id

-- process orderlines

                while (@li_no < @o_ol_cnt)
                begin

                                select @li_no = @li_no + 1


-- set i_id, s_w_id, and qty for this lineitem

                                select          @li_id = case @li_no
                                                        when 1 then @i_id1
                                                        when 2 then @i_id2
                                                        when 3 then @i_id3
                                                        when 4 then @i_id4
                                                        when 5 then @i_id5
                                                        when 6 then @i_id6
                                                        when 7 then @i_id7
                                                        when 8 then @i_id8
                                                        when 9 then @i_id9
                                                        when 10 then @i_id10
                                                        when 11 then @i_id11
                                                        when 12 then @i_id12
                                                        when 13 then @i_id13
                                                        when 14 then @i_id14
                                                        when 15 then @i_id15
                                                        end,

                                @li_s_w_id = case @li_no
                                                        when 1 then
@s_w_id1
                                                        when 2 then
@s_w_id2
                                                        when 3 then
@s_w_id3
                                                        when 4 then
@s_w_id4
                                                        when 5 then
@s_w_id5
                                                        when 6 then
@s_w_id6
                                                        when 7 then
@s_w_id7
                                                        when 8 then
@s_w_id8
                                                        when 9 then
@s_w_id9
                                                        when 10 then
@s_w_id10
                                                        when 11 then
@s_w_id11
                                                        when 12 then
@s_w_id12
                                                        when 13 then
@s_w_id13
                                                        when 14 then
@s_w_id14
                                                        when 15 then
@s_w_id15
                                                        end,

                                @li_qty = case  @li_no
                                                        when 1 then
@ol_qty1
                                                        when 2 then
@ol_qty2
                                                        when 3 then
@ol_qty3
                                                        when 4 then
@ol_qty4
                                                        when 5 then
@ol_qty5
                                                        when 6 then
@ol_qty6
                                                        when 7 then
@ol_qty7
                                                        when 8 then
@ol_qty8
                                                        when 9 then
@ol_qty9
                                                        when 10 then
@ol_qty10
                                                        when 11 then
@ol_qty11
                                                        when 12 then
@ol_qty12
                                                        when 13 then
@ol_qty13
                                                        when 14 then
@ol_qty14
                                                        when 15 then
@ol_qty15
                                                        end

-- get item data (no one updates item)

                                select          @i_price = i_price,
                                                @i_name  = i_name,
                                                @i_data  = i_data
                                from            item (tablock repeatableread)
                                where           i_id = @li_id

-- update stock values

                                update          stock
                                set             s_ytd                   = s_ytd
+ @li_qty,
                                                @s_quantity  = s_quantity   =
s_quantity - @li_qty +

                        case when (s_quantity - @li_qty < 10) then 91 else 0 end,
                                                s_order_cnt   = s_order_cnt + 1,
                                                s_remote_cnt                =
s_remote_cnt + case when (@li_s_w_id = @w_id) then 0 else 1 end,
                                                @s_data                 = s_data,
                                                @s_dist                 = case
@d_id
                                                                        when 1
then s_dist_01
                                                                        when 2
then s_dist_02
                                                                        when 3
then s_dist_03
                                                                        when 4
then s_dist_04
```

```
                                                        when 5
then s_dist_05
                                                        when 6
then s_dist_06
                                                        when 7
then s_dist_07
                                                        when 8
then s_dist_08
                                                        when 9
then s_dist_09
                                                        when
10 then s_dist_10
                                                        end
                        where   s_i_id                  = @li_id
and
                                s_w_id                  =
@li_s_w_id

-- if there actually is a stock (and item) with these ids, go to work

                        if (@@rowcount > 0)
                        begin

-- insert order_line data (using data from item and stock)

                                insert into order_line values(@o_id,

                @d_id,

                @w_id,

                @li_no,

                @li_id,

                @li_s_w_id,

                'dec 31, 1899',

                @li_qty,

                @i_price * @li_qty,

                @s_dist)

-- send line-item data to client

                                select  @i_name,
                                        @s_quantity,
                                        b_g = case when (
(patindex('%ORIGINAL%',@i_data) > 0) and
                (patindex('%ORIGINAL%',@s_data) > 0) ) then 'B' else 'G'
end,
                                        @i_price,
                                        @i_price * @li_qty

                        end
                        else
                        begin

-- no item (or stock) found - triggers rollback condition

                                select ",0,",0,0
                                select @commit_flag = 0

                        end
                end

-- get customer last name, discount, and credit rating

                select  @c_last     = c_last,
                        @c_discount = c_discount,
                        @c_credit   = c_credit,
                        @c_id_local = c_id
                from    customer (repeatableread)
                where   c_id                    = @c_id and
                        c_w_id                  = @w_id and
                        c_d_id                  = @d_id

-- insert fresh row into orders table

                insert into orders values (  @o_id,
                                                        @d_id,
                                                        @w_id,

                @c_id_local,

                @o_entry_d,
                                                        0,

                @o_ol_cnt,

                @o_all_local)

-- insert corresponding row into new-order table

                insert into new_order values (          @o_id,
                                                        @d_id,
                                                        @w_id)

-- select warehouse tax

                select  @w_tax      = w_tax
                from    warehouse (repeatableread)
                where   w_id        = @w_id

                if (@commit_flag = 1)

                        commit transaction n
                else

-- all that work for nuthin!!!

                        rollback transaction n

-- return order data to client

                select  @w_tax,
                        @d_tax,
                        @o_id,
                        @c_last,
                        @c_discount,
                        @c_credit,
                        @o_entry_d,
                        @commit_flag

end

go
```

ORDSTAT.SQL

```
-- File:    ORDSTAT.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20.000
--          Copyright Microsoft, 1999
-- Purpose:  Creates order status transaction stored procedure
--
--              Interface Level: 4.10.000

use tpcc
go

if exists ( select name from sysobjects where name = "tpcc_orderstatus" )
                drop procedure   tpcc_orderstatus
go

create proc tpcc_orderstatus            @w_id       smallint,
                                        @d_id       tinyint,

                                        @c_id       int,
```

```
                                        @c_last       char(16)
= ""


as

declare @c_balance              numeric(12,2),
        @c_first                char(16),
        @c_middle               char(2),
        @o_id                   int,
        @o_entry_d              datetime,
        @o_carrier_id           smallint,
        @cnt                    smallint

begin tran o

if (@c_id = 0)
                begin

-- get customer id and info using last name

                        select  @cnt          = (count(*)+1)/2
                        from    customer (repeatableread)
                        where   c_last        = @c_last and
                                c_w_id        = @w_id and
                                c_d_id        = @d_id

                        set     rowcount @cnt

                        select  @c_id                   = c_id,
                                @c_balance  = c_balance,
                                  @c_first    = c_first,
                                  @c_last     = c_last,
                                  @c_middle = c_middle
                        from    customer (repeatableread)
                        where   c_last                  =
@c_last and
                                c_w_id                  = @w_id
and
                                c_d_id                  = @d_id
                        order   by c_w_id, c_d_id, c_last, c_first

                        set     rowcount 0
                end

                else

                begin

-- get customer info if by id

                        select  @c_balance  = c_balance,
                                @c_middle   = c_middle,
                                @c_first    = c_first,
                                @c_last                 = c_last
                        from    customer (repeatableread)
                        where   c_id                    = @c_id
and
                                c_d_id                  = @d_id
and
                                c_w_id                  = @w_id


                        select  @cnt          = @@rowcount

                end

-- if no such customer

                if (@cnt = 0)
                begin
                        raiserror("Customer not found",18,1)
                        goto custnotfound
                end

-- get order info

                select  @o_id                   = o_id,
                        @o_entry_d  = o_entry_d,
                        @o_carrier_id           = o_carrier_id

                from    orders (serializable)
                where   o_c_id                  = @c_id and
                        o_d_id                  = @d_id and
                        o_w_id                  = @w_id
                order   by o_id asc

-- select order lines for the current order

                select  ol_supply_w_id,
                        ol_i_id,
                        ol_quantity,
                        ol_amount,
                        ol_delivery_d
                from    order_line (repeatableread)
                where   ol_o_id = @o_id and
                        ol_d_id = @d_id and
                        ol_w_id = @w_id

custnotfound:

commit tran o

-- return data to client

select  @c_id,
        @c_first,
        @c_last,
        @c_middle,
        @o_entry_d,
        @o_carrier_id,
        @c_balance,
        @o_id

go
```

DELIVERY.SQL

```
-- File:    DELIVERY.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20.000
--          Copyright Microsoft, 1999
-- Purpose:  Creates delivery transaction stored procedure
--
--              Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_delivery" )
                drop procedure tpcc_delivery
go

create proc tpcc_delivery   @w_id          smallint,

                                                        @o_carrier_id
smallint
as


declare @d_id               tinyint,
        @o_id  int,
        @c_id  int,
        @total  numeric(12,2),
        @oid1  int,
        @oid2  int,
        @oid3  int,
        @oid4  int,
        @oid5  int,
        @oid6  int,
        @oid7  int,
        @oid8  int,
        @oid9  int,
        @oid10 int

select @d_id = 0
```

```
begin tran d

        while (@d_id < 10)
        begin

            select  @d_id  = @d_id + 1,
                            @total = 0,
                            @o_id  = 0

                select      top 1
                            @o_id       = no_o_id
                from        new_order (serializable updlock)
                where       no_w_id     = @w_id and
                            no_d_id     = @d_id
                order       by no_o_id asc

                if (@@rowcount <> 0)
                    begin

-- claim the order for this district

                                delete      new_order
                                where       no_w_id     = @w_id
and
                                            no_d_id     = @d_id
and
                                            no_o_id     = @o_id

-- set carrier_id on this order (and get customer id)

                                update      orders
                                set         o_carrier_id  =
@o_carrier_id,
                                            @c_id
        = o_c_id
                                where       o_w_id
        = @w_id and
                                            o_d_id
        = @d_id and
                                            o_id
        = @o_id

-- set date in all lineitems for this order (and sum amounts)

                                update      order_line
                                set         ol_delivery_d =
getdate(),
                                            @total
        = @total + ol_amount
                                where       ol_w_id
        = @w_id and
                                            ol_d_id
        = @d_id and
                                            ol_o_id
        = @o_id

-- accummulate lineitem amounts for this order into customer

                                update      customer
                                set         c_balance     =
c_balance + @total,
                                            c_delivery_cnt
        = c_delivery_cnt + 1
                                where       c_w_id
        = @w_id and
                                            c_d_id
        = @d_id and
                                            c_id
        = @c_id

                    end

            select @oid1 = case @d_id when  1  then @o_id else @oid1
end,
            @oid2 = case @d_id when  2  then @o_id else @oid2 end,
            @oid3 = case @d_id when  3  then @o_id else @oid3 end,
            @oid4 = case @d_id when  4  then @o_id else @oid4 end,
            @oid5 = case @d_id when  5  then @o_id else @oid5 end,
            @oid6 = case @d_id when  6  then @o_id else @oid6 end,
            @oid7 = case @d_id when  7  then @o_id else @oid7 end,
            @oid8 = case @d_id when  8  then @o_id else @oid8 end,
            @oid9 = case @d_id when  9  then @o_id else @oid9 end,
            @oid10 = case @d_id when 10 then @o_id else @oid10 end

    end

commit tran d

-- return delivery data to client

select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

go
```

PAYMENT.SQL

```
-- File:     PAYMENT.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.41
--           Copyright Microsoft, 2001
-- Purpose:  Creates payment transaction stored procedure
--
--               Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = 'tpcc_payment' )
            drop procedure tpcc_payment
go

create proc tpcc_payment  @w_id          smallint,
                                                    @c_w_id
smallint,
                                                    @h_amount
numeric(6,2),
                                                    @d_id          tinyint,
                                                    @c_d_id        tinyint,
                                                    @c_id          int,
                                                    @c_last        char(16)
= ''




as
declare  @w_street_1     char(20),
        @w_street_2      char(20),
        @w_city          char(20),
        @w_state         char(2),
        @w_zip           char(9),
        @w_name          char(10),
        @d_street_1      char(20),
        @d_street_2      char(20),
        @d_city          char(20),
        @d_state         char(2),
        @d_zip           char(9),
        @d_name          char(10),
        @c_first         char(16),
        @c_middle        char(2),
        @c_street_1      char(20),
        @c_street_2      char(20),
        @c_city          char(20),
        @c_state         char(2),
        @c_zip           char(9),
        @c_phone         char(16),
        @c_since         datetime,
```

```
        @c_credit       char(2),
        @c_credit_lim   numeric(12,2),
        @c_balance      numeric(12,2),
        @c_discount     numeric(4,4),
        @data           char(500),
        @c_data         char(500),
        @datetime       datetime,
        @w_ytd          numeric(12,2),
        @d_ytd          numeric(12,2),
        @cnt            smallint,
        @val            smallint,
        @screen_data            char(200),
                @d_id_local  tinyint,
                @w_id_local  smallint,
                @c_id_local  int

select @screen_data = ''

begin tran p

-- get payment date

                select          @datetime = getdate()

                if (@c_id = 0)
                begin

-- get customer id and info using last name

                                select          @cnt            = count(*)
                                from            customer (repeatableread)
                                where           c_last          = @c_last and
                                                c_w_id          = @c_w_id and
                                                c_d_id          = @c_d_id



                                select          @val = (@cnt + 1) / 2
                                set             rowcount @val

                                select          @c_id           = c_id
                                from            customer (repeatableread)
                                where           c_last          = @c_last and
                                                c_w_id          = @c_w_id and
                                                c_d_id          = @c_d_id
                                order           by c_last, c_first

                                set             rowcount 0
                end

-- get customer info and update balances

                update          customer
                set             @c_balance      = c_balance - @h_amount,
                                c_payment_cnt           = c_payment_cnt + 1,
                                c_ytd_payment           = c_ytd_payment + @h_amount,
                                @c_first        = c_first,
                                @c_middle       = c_middle,
                                @c_last                 = c_last,
                                @c_street_1     = c_street_1,
                                @c_street_2     = c_street_2,
                                @c_city                 = c_city,
                                @c_state        = c_state,
                                @c_zip                  = c_zip,
                                @c_phone        = c_phone,
                                @c_credit       = c_credit,
                                @c_credit_lim           = c_credit_lim,
                                @c_discount     = c_discount,
                                @c_since        = c_since,
                                @data                   = c_data,

                                @c_id_local     = c_id
                where           c_id                    = @c_id and
                                c_w_id                  = @c_w_id and
                                c_d_id                  = @c_d_id

-- if customer has bad credit get some more info

                if (@c_credit = 'BC')
                begin

--              compute new info

                                select @c_data          =
convert(char(5),@c_id) +

convert(char(4),@c_d_id) +

convert(char(5),@c_w_id) +

convert(char(4),@d_id) +

convert(char(5),@w_id) +

convert(char(19),@h_amount) +
                                                substring(@data, 1,
458)

-- update customer info

                                update          customer
                                set             c_data          = @c_data
                                where           c_id            = @c_id and
                                                c_w_id          = @c_w_id and
                                                c_d_id          = @c_d_id

                                select          @screen_data = substring
(@c_data,1,200)
                end

-- get district data and update year-to-date

                update          district
                set             d_ytd                   = d_ytd +
@h_amount,
                                @d_street_1     = d_street_1,
                                @d_street_2     = d_street_2,
                                @d_city                 = d_city,
                                @d_state        = d_state,
                                @d_zip                  = d_zip,
                                @d_name                 = d_name,
                                @d_id_local     = d_id
                where           d_w_id                  = @w_id and
                                d_id                    = @d_id

-- get warehouse data and update year-to-date

                update          warehouse
                set             w_ytd                   = w_ytd +
@h_amount,
                                @w_street_1     = w_street_1,
                                @w_street_2     = w_street_2,
                                @w_city                 = w_city,
                                @w_state        = w_state,
                                @w_zip                  = w_zip,
                                @w_name                 = w_name,
                                @w_id_local     = w_id
                where           w_id                    = @w_id

-- create history record

                insert into history values (  @c_id_local,
                                                                @c_d_id,
                                                                @c_w_id,

                @d_id_local,

                @w_id_local,

                @datetime,

                @h_amount,

                @w_name + '    ' + @d_name)
commit tran p
```

```
-- return data to client

select          @c_id,
                @c_last,
                @datetime,
                @w_street_1,
                @w_street_2,
                @w_city,
                @w_state,
                @w_zip,
                @d_street_1,
                @d_street_2,
                @d_city,
                @d_state,
                @d_zip,
                @c_first,
                @c_middle,
                @c_street_1,
                @c_street_2,
                @c_city,
                @c_state,
                @c_zip,
                @c_phone,
                @c_since,
                @c_credit,
                @c_credit_lim,
                @c_discount,
                @c_balance,
                @screen_data

go
```

STOCKLEV.SQL

```
-- File:    STOCKLEV.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.41
--          Copyright Microsoft, 2001
-- Purpose:  Creates stock level transaction stored procedure
--
--              Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = 'tpcc_stocklevel' )
                drop procedure tpcc_stocklevel
go

create proc tpcc_stocklevel@w_id          smallint,
                                                @d_id           tinyint,
                                                @threshhold smallint
as

declare         @o_id_low int,
                @o_id_high int

select          @o_id_low    = (d_next_o_id - 20),
                @o_id_high   = (d_next_o_id - 1)
from            district
where           d_w_id                  = @w_id and
                d_id                    = @d_id

select          count(distinct(s_i_id))
from            stock, order_line
where           ol_w_id                 =  @w_id and
                ol_d_id                 =  @d_id and
                ol_o_id                   between @o_id_low and
                                          @o_id_high and
                s_w_id                  =  ol_w_id and
                s_i_id                  =  ol_i_id and
                s_quantity      <  @threshhold

go
```

TPCC.H

```
//              File:                   TPCC.H
//                                                      Microsoft TPC-C Kit
Ver. 4.41
//                                                      Copyright Microsoft,
1996, 1997, 1998, 1999, 2000, 2001
//              Purpose:        Header file for TPC-C database loader


// Build number of TPC Benchmark Kit
#define TPCKIT_VER      "4.41"

// 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
"C:\\MSTPCC.440\\SETUP\\logs\\load.out"
#define LOG_PATH
"C:\\MSTPCC.440\\SETUP\\LOGS\\";
#define LOADER_NURAND_C                                         123
#define DEF_STARTING_WAREHOUSE          1
#define BUILD_INDEX
                1               // build both data and indexes
#define INDEX_ORDER
                1               // build indexes before load
#define SCALE_DOWN
                0   // build a normal scale database
#define INDEX_SCRIPT_PATH               "scripts"

typedef struct
{
    char                                        *server;
    char                                        *database;
    char                                        *user;
    char                                        *password;
                BOOL
                tables_all;                     // set if loading all
tables
                BOOL
                table_item;                     // set if loading ITEM
table specifically
                BOOL
                table_warehouse;        // set if loading WAREHOUSE,
DISTRICT, and STOCK
```

```
        BOOL
        table_customer;                 // set if loading
CUSTOMER and HISTORY
        BOOL
        table_orders;           // set if loading NEW-ORDER,
ORDERS, ORDER-LINE
   long                                 num_warehouses;
   long                                 batch;
   long                                 verbose;
        long
        pack_size;
        char
        *loader_res_file;
        char
        *log_path;
        char
        *synch_servername;
        long
        case_sensitivity;
        long
        starting_warehouse;
        long
        build_index;
        long
        index_order;
        long
        scale_down;
        char
        *index_script_path;
} TPCCLDR_ARGS;

// String length constants
        #define SERVER_NAME_LEN         20
        #define DATABASE_NAME_LEN       20
        #define USER_NAME_LEN           20
        #define PASSWORD_LEN            20
        #define TABLE_NAME_LEN          20
        #define I_DATA_LEN              50
        #define I_NAME_LEN              24
        #define BRAND_LEN               1
        #define LAST_NAME_LEN           16
        #define W_NAME_LEN              10
        #define ADDRESS_LEN             20
        #define STATE_LEN               2
        #define ZIP_LEN                 9
        #define S_DIST_LEN              24
        #define S_DATA_LEN              50
        #define D_NAME_LEN              10
        #define FIRST_NAME_LEN          16
        #define MIDDLE_NAME_LEN         2
        #define PHONE_LEN               16
        #define CREDIT_LEN              2
        #define H_DATA_LEN              24
        #define C_DATA_LEN              500
        #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. 4.41
//                                          Copyright Microsoft,
1996, 1997, 1998, 1999, 2000, 2001
//          Purpose:    Source file for TPC-C database loader

// Includes
#include "tpcc.h"
#include "search.h"

// Defines
#define MAXITEMS            100000
#define MAXITEMS_SCALE_DOWN                 100
#define CUSTOMERS_PER_DISTRICT  3000
#define CUSTOMERS_SCALE_DOWN    30
#define DISTRICT_PER_WAREHOUSE  10
#define ORDERS_PER_DISTRICT     3000
#define ORDERS_SCALE_DOWN       30
#define MAX_CUSTOMER_THREADS    2
#define MAX_ORDER_THREADS       3
#define MAX_MAIN_THREADS        4

// Functions declarations

void HandleErrorDBC (SQLHDBC  hdbc1);

void 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      short                                   c_ytd_payment;
                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           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];
                char                    err_log_path[256];

                // 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);
                strcpy(err_log_path,aptr->log_path);
                strcat(err_log_path,"item.err");
                rc = bcp_init(i_hdbc1, name, NULL, err_log_path , DB_IN);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                {
                        sprintf(bcphint, "tablock, order (i_id),
ROWS_PER_BATCH = 100000");
                        rc = bcp_control(i_hdbc1, BCPHINTS, (void*)
bcphint);
                        if (rc != SUCCEED)
                                HandleErrorDBC(i_hdbc1);
                }
                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];
	char		err_log_path[256];

	// Seed with unique number
	seed(2);

	printf("Loading warehouse table...\n");

	// if build index before load...
	if ((aptr->build_index == 1) && (aptr->index_order == 1))
		BuildIndex("idxwarcl");
	InitString(w_name, W_NAME_LEN+1);
	InitAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

	sprintf(name, "%s..%s", aptr->database, "warehouse");

	//rc = bcp_init(w_hdbc1, name, NULL, "logs\\whouse.err",
DB_IN);
	strcpy(err_log_path,aptr->log_path);
	strcat(err_log_path,"whouse.err");
	rc = bcp_init(w_hdbc1, name, NULL, err_log_path, DB_IN);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	if ((aptr->build_index == 1) && (aptr->index_order == 1))
	{
		sprintf(bcphint, "tablock, order (w_id),
ROWS_PER_BATCH = %d", aptr->num_warehouses);
		rc = bcp_control(w_hdbc1, BCPHINTS, (void*)
bcphint);
		if (rc != SUCCEED)
			HandleErrorDBC(w_hdbc1);
	}

	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];
            char        err_log_path[256];

            // Seed with unique number
            seed(4);

            printf("Loading district table...\n");

            // build index before load
            if ((aptr->build_index == 1) && (aptr->index_order == 1))
                    BuildIndex("idxdiscl");

            InitString(d_name, D_NAME_LEN+1);
            InitAddress(d_street_1, d_street_2, d_city, d_state, d_zip);
            sprintf(name, "%s..%s", aptr->database, "district");

            //rc = bcp_init(w_hdbc1, name, NULL, "logs\\district.err",
DB_IN);     strcpy(err_log_path,aptr->log_path);
            strcat(err_log_path,"district.err");
            rc = bcp_init(w_hdbc1, name, NULL, err_log_path, DB_IN);
            if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

            if ((aptr->build_index == 1) && (aptr->index_order == 1))
            {
                    sprintf(bcphint, "tablock, order (d_w_id, d_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 10));
                    rc = bcp_control(w_hdbc1, BCPHINTS, (void*)
bcphint);
                    if (rc != SUCCEED)
                            HandleErrorDBC(w_hdbc1);
            }

            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];
		char		err_log_path[256];

		// Seed with unique number
		seed(3);

		// if build index before load...
		if ((aptr->build_index == 1) && (aptr->index_order == 1))
			BuildIndex("idxstkcl");

		sprintf(name, "%s..%s", aptr->database, "stock");

		//rc = bcp_init(w_hdbc1, name, NULL, "logs\\stock.err",
DB_IN);
		strcpy(err_log_path,aptr->log_path);
		strcat(err_log_path,"stock.err");
		rc = bcp_init(w_hdbc1, name, NULL, err_log_path, DB_IN);
		if (rc != SUCCEED)
			HandleErrorDBC(w_hdbc1);

		if ((aptr->build_index == 1) && (aptr->index_order == 1))
		{
			sprintf(bcphint, "tablock, order (s_i_id, s_w_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 100000));
			rc = bcp_control(w_hdbc1, BCPHINTS, (void*)
bcphint);
			if (rc != SUCCEED)
				HandleErrorDBC(w_hdbc1);
		}

		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];
        int
                        num_procs;
        char
        err_log_path_cust[256];
        char
        err_log_path_hist[256];

        // SQLRETURN
        rc_t;
        // 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");
                        // check the number of processors on this system
                        // if 8 or more processors, then build index on
History.
                        // if less than 8 processors, do not build the index
                        num_procs = atoi(getenv(
"NUMBER_OF_PROCESSORS" ));
                        if ( num_procs >= 8 )
                                BuildIndex("idxhiscl");
        }

        // Initialize bulk copy
        sprintf(name, "%s..%s", aptr->database, "customer");

        //rc = bcp_init(c_hdbc1, name, NULL, "logs\\customer.err",
DB_IN);
        strcpy(err_log_path_cust,aptr->log_path);
        strcat(err_log_path_cust,"customer.err");
        rc = bcp_init(c_hdbc1, name, NULL, err_log_path_cust,
DB_IN);
        if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                        sprintf(bcphint, "tablock, order (c_w_id, c_d_id,
c_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
                        rc = bcp_control(c_hdbc1, BCPHINTS, (void*)
bcphint);
                        if (rc != SUCCEED)
                                HandleErrorDBC(c_hdbc1);
        }

        sprintf(name, "%s..%s", aptr->database, "history");

        rc = bcp_init(c_hdbc2, name, NULL, "logs\\history.err", DB_IN);
        strcpy(err_log_path_hist,aptr->log_path);
```

```
		strcat(err_log_path_hist,"history.err");
		rc = bcp_init(c_hdbc2, name, NULL, err_log_path_hist, DB_IN);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc2);

		sprintf(bcphint, "tablock");
		rc = bcp_control(c_hdbc2, BCPHINTS, (void*) bcphint);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc2);

		customer_rows_loaded    = 0;
		history_rows_loaded     = 0;

		CustomerBufInit();

		customer_time_start.time_start = (TimeNow() / MILLI);
		history_time_start.time_start = (TimeNow() / MILLI);

		for (w_id = (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 bcp connection
		rcint = bcp_done(c_hdbc1);
		if (rcint < 0)
			HandleErrorDBC(c_hdbc1);

		rcint = bcp_done(c_hdbc2);
		if (rcint < 0)
			HandleErrorDBC(c_hdbc2);

		printf("Finished loading customer table.\n");

		// if build index after load...
		if ((aptr->build_index == 1) && (aptr->index_order == 0))
		{
			BuildIndex("idxcuscl");
			// check the number of processors on this system
			// if 8 or more processors, then build index on
History.
			// if less than 8 processors, do not build the index
			num_procs = atoi(getenv(
"NUMBER_OF_PROCESSORS" ));
			if (num_procs >= 8)
				BuildIndex("idxhiscl");
		}

		// build non-clustered index
		if (aptr->build_index == 1)
			BuildIndex("idxcusnc");

		// Output the NURAND used for the loader into C_FIRST for
C_ID = 1,
		// C_W_ID = 1, and C_D_ID = 1
		//sprintf(cmd, "osql -S%s -U%s -P%s -d%s -e -Q\"update
customer set c_first = 'C_LOAD = %d' where c_id = 1 and c_w_id = 1 and
c_d_id = 1\" > logs\\nurand_load.log",
		sprintf(cmd, "osql -S%s -U%s -P%s -d%s -e -Q\"update
customer set c_first = 'C_LOAD = %d' where c_id = 1 and c_w_id = 1 and
c_d_id = 1\" > %snurand_load.log",
					aptr->server,
					aptr->user,
					aptr->password,
					aptr->database,
					LOADER_NURAND_C,
					aptr->log_path);

		system(cmd);

		SQLFreeStmt(c_hstmt1, SQL_DROP);
		SQLDisconnect(c_hdbc1);
		SQLFreeConnect(c_hdbc1);

		SQLFreeStmt(c_hstmt2, SQL_DROP);
		SQLDisconnect(c_hdbc2);
		SQLFreeConnect(c_hdbc2);
```

```
    return;

}


//================================================
=========================
//
// Function   : CustomerBufInit
//
//================================================
=========================

void CustomerBufInit()
{
        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];
                char
                err_log_path_ord[256];
                char
                err_log_path_nord[256];
                char
                err_log_path_ordl[256];

                // seed with unique number
                seed(6);
```

```
	printf("Loading orders...\n");

	// if build index before load...
	if ((aptr->build_index == 1) && (aptr->index_order == 1))
	{
				BuildIndex("idxordcl");
				BuildIndex("idxnodcl");
				BuildIndex("idxodlcl");
	}

	// initialize bulk copy
	sprintf(name, "%s..%s", aptr->database, "orders");

	rc = bcp_init(o_hdbc1, name, NULL, "logs\\orders.err", DB_IN);
	strcpy(err_log_path_ord,aptr->log_path);
	strcat(err_log_path_ord,"orders.err");
	rc = bcp_init(o_hdbc1, name, NULL, err_log_path_ord, DB_IN);
	if (rc != SUCCEED)
		HandleErrorDBC(o_hdbc1);

	if ((aptr->build_index == 1) && (aptr->index_order == 1))
	{
		sprintf(bcphint, "tablock, order (o_w_id, o_d_id,
o_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
		rc = bcp_control(o_hdbc1, BCPHINTS, (void*)
bcphint);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc1);
	}

	sprintf(name, "%s..%s", aptr->database, "new_order");

	rc = bcp_init(o_hdbc2, name, NULL, "logs\\neword.err",
DB_IN);
	strcpy(err_log_path_nord,aptr->log_path);
	strcat(err_log_path_nord,"neword.err");
	rc = bcp_init(o_hdbc2, name, NULL, err_log_path_nord,
DB_IN);
	if (rc != SUCCEED)
		HandleErrorDBC(o_hdbc2);

	if ((aptr->build_index == 1) && (aptr->index_order == 1))
	{
		sprintf(bcphint, "tablock, order (no_w_id, no_d_id,
no_o_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 9000));
		rc = bcp_control(o_hdbc2, BCPHINTS, (void*)
bcphint);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc2);
	}

	sprintf(name, "%s..%s", aptr->database, "order_line");

	rc = bcp_init(o_hdbc3, name, NULL, "logs\\ordline.err",
DB_IN);
	strcpy(err_log_path_ordl,aptr->log_path);
	strcat(err_log_path_ordl,"ordline.err");
	rc = bcp_init(o_hdbc3, name, NULL, err_log_path_ordl,
DB_IN);
	if (rc != SUCCEED)
		HandleErrorDBC(o_hdbc3);

   if ((aptr->build_index == 1) && (aptr->index_order == 1))
	{
		sprintf(bcphint, "tablock, order (ol_w_id, ol_d_id,
ol_o_id, ol_number), ROWS_PER_BATCH = %u", (aptr->num_warehouses *
300000));
		rc = bcp_control(o_hdbc3, BCPHINTS, (void*)
bcphint);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc3);
	}

	orders_rows_loaded      = 0;
	new_order_rows_loaded   = 0;
	order_line_rows_loaded  = 0;

	OrdersBufInit();

	orders_time_start.time_start = (TimeNow() / MILLI);
	new_order_time_start.time_start = (TimeNow() / MILLI);
	order_line_time_start.time_start = (TimeNow() / MILLI);

	for (w_id = (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);
```

```
                        SQLDisconnect(o_hdbc2);
                        SQLFreeStmt(o_hstmt2, SQL_DROP);
                        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",
				table_name,
				aptr->batch,
				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, "osql -S%s -U%s -P%s -e -i%s\\%s.sql >
%s%s.log",
                                        aptr->server,
                                        aptr->user,
                                        aptr->password,
                                        aptr-
>index_script_path,
                                        index_script,
                                        aptr->log_path,
                                        index_script);

        system(cmd);

        printf("Finished index creation:  %s\n",index_script);
}


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];
        char                err_log_path[256];
        FILE                *fp1;

        i = 1;
        while (( rc2 = SQLGetDiagRec(SQL_HANDLE_DBC , hdbc1, i,
SqlState , &NativeError,
                                        Msg, sizeof(Msg) ,
&MsgLen )) != SQL_NO_DATA )
        {

                sprintf( szLastError , "%s" ,  Msg );
```

```
                        _strtime(timebuf);
                        _strdate(datebuf);

                        printf( "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);

                        strcpy(err_log_path,aptr->log_path);
                        strcat(err_log_path,"tpccldr.err");
                        fp1 = fopen(err_log_path,"w");
                        //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];
                char                    err_log_path[256];
                FILE                    *fp1;

                i = 1;
                while (( rc2 = SQLGetDiagRec(SQL_HANDLE_STMT , hstmt1, i,
SqlState , &NativeError,
                                                Msg, sizeof(Msg) ,
&MsgLen )) != SQL_NO_DATA )
                {

                        sprintf( szLastError , "%s" ,  Msg );

                        _strtime(timebuf);
                        _strdate(datebuf);

                        printf( "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);

                        strcpy(err_log_path,aptr->log_path);
                        strcat(err_log_path,"tpccldr.err");
                        fp1 = fopen(err_log_path,"w");
                        //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;
}
```

RANDOM.C

```
//              File:                   RANDOM.C
//                                                      Microsoft TPC-C Kit
Ver. 4.41
//                                                      Copyright Microsoft,
1996, 1997, 1998, 1999, 2000, 2001
//              Purpose:        Random number generation routines for database
loader


// Includes
#include "tpcc.h"
#include "math.h"

// Defines
#define A           16807
#define M       2147483647
#define Q           127773      /* M div A */
#define R            2836       /* M mod A */
#define Thread              __declspec(thread)

// Globals
long            Thread Seed = 0;        /* thread local seed */
/*****************************************************************
*****************
*                                                               *
* random -                                                      *
*       Implements a GOOD pseudo random number generator.  This
generator     *
*       will/should? run the complete period before repeating.          *
*                                                               *
* Copied from:                                                  *
*       Random Numbers Generators: Good Ones Are Hard to Find.
*
*       Communications of the ACM - October 1988 Volume 31 Number 10
*
*                                                               *
* Machine Dependencies:                                         *
*       long must be 2 ^ 31 - 1 or greater.                     *
*                                                               *
*****************************************************************
***************/

/*****************************************************************
*****************
* seed - load the Seed value used in irand and drand.  Should be used before
*
*       first call to irand or drand.                           *
*****************************************************************
***************/

void seed(long val)
{

#ifdef DEBUG
   printf("[%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;
}
```

GETARGS.C

```
//              File:                   GETARGS.C
//                                                      Microsoft TPC-C Kit
Ver. 4.41
//                                                      Copyright Microsoft,
1996, 1997, 1998, 1999, 2000, 2001
//              Purpose:        Source file for command line processing


// Includes
#include "tpcc.h"

//=================================================
=========================
//
// Function name: GetArgsLoader
//
//=================================================
=========================

void GetArgsLoader(int argc, char **argv, TPCCLDR_ARGS *pargs)
{
   int                  i;
   char         *ptr;

#ifdef DEBUG
   printf("[%ld]DBG: Entering GetArgsLoader()\n", (int)
GetCurrentThreadId());
#endif

   /* init args struct with some useful values */
   pargs->server                        = SERVER;
   pargs->user                          = USER;
   pargs->password                      = PASSWORD;
   pargs->database                      = DATABASE;
   pargs->batch                         = BATCH;
   pargs->num_warehouses                        = UNDEF;
        pargs->tables_all                               = TRUE;
        pargs->table_item                               =
FALSE;
        pargs->table_warehouse          = FALSE;
        pargs->table_customer           = FALSE;
        pargs->table_orders                             =
FALSE;
        pargs->loader_res_file          = LOADER_RES_FILE;
        pargs->log_path
        = LOG_PATH;
        pargs->pack_size                = DEFLDPACKSIZE;
        pargs->starting_warehouse       =
DEF_STARTING_WAREHOUSE;
        pargs->build_index                              =
BUILD_INDEX;
        pargs->index_order                              =
INDEX_ORDER;
        pargs->index_script_path  =  INDEX_SCRIPT_PATH;
        pargs->scale_down                               =
SCALE_DOWN;

   /* check for zero command line args */
   if ( argc == 1 )
                GetArgsLoaderUsage();

   for (i = 1; i < argc; ++i)
   {
                if (argv[i][0] != '-' && argv[i][0] != '/')
                {
                printf("\nUnrecognized command");
                GetArgsLoaderUsage();
                exit(1);
                }

                ptr = argv[i];

                switch (ptr[1])
                {
                case '?':    /* Fall throught */
                                GetArgsLoaderUsage();
                                break;

                case 'D':
                                pargs->database =
ptr+2;
                                break;

                case 'P':
                                pargs->password =
ptr+2;
                                break;

                case 'S':
                                pargs->server =
ptr+2;
                                break;

                case 'U':
                                pargs->user = ptr+2;
                                break;

                case 'b':
                                pargs->batch =
atol(ptr+2);
                                break;

                case 'W':               pargs-
>num_warehouses = atol(ptr+2);
                                break;

                case 's':
                                pargs-
>starting_warehouse = atol(ptr+2);
                                break;

                case 't':
                                {
                                        pargs-
>tables_all = FALSE;
                                        if
(strcmp(ptr+2,"item") == 0)

        pargs->table_item = TRUE;
                                        else if
(strcmp(ptr+2,"warehouse") == 0)

        pargs->table_warehouse = TRUE;
                                        else if
(strcmp(ptr+2,"customer") == 0)

        pargs->table_customer = TRUE;
                                        else if
(strcmp(ptr+2,"orders") == 0)

        pargs->table_orders = TRUE;
                                        else
                                        {

        printf("\nUnrecognized command");

        GetArgsLoaderUsage();

                                        exit(1);
                                        }

                                        break;
                                }
                                case 'f':
                                pargs->loader_res_file
= ptr+2;
                                break;

                                case 'L':
                                pargs->log_path =
ptr+2;
                                break;

                                case 'p':
```

```
                                        pargs->pack_size =
atol(ptr+2);
                                        break;

                                case 'i':
                                        pargs->build_index =
atol(ptr+2);
                                        break;

                                case 'o':
                                        pargs->index_order =
atol(ptr+2);
                                        break;

                                case 'c':
                                        pargs->scale_down =
atol(ptr+2);
                                        break;

                        case 'd':
                                        pargs-
>index_script_path = ptr+2;
                                        break;

                                default:
                                GetArgsLoaderUsage();
                                exit(-1);
                                break;
                        }
    }

    /* check for required args */
    if (pargs->num_warehouses == UNDEF )
    {
                        printf("Number of Warehouses is required\n");
                        exit(-2);
    }

    return;
}


//=============================================
========================
//
// Function name: GetArgsLoaderUsage
//
//=============================================
========================

void GetArgsLoaderUsage()
{

#ifdef DEBUG
    printf("[%ld]DBG: Entering GetArgsLoaderUsage()\n", (int)
GetCurrentThreadId());
#endif


    printf("TPCCLDR:\n\n");
                printf("Parameter                                       Default\n");
                printf("-------------------------------------------------------------\n");
    printf("-W Number of Warehouses to Load                 Required \n");
    printf("-S Server                                   %s\n", SERVER);
    printf("-U Username                                   %s\n", USER);
    printf("-P Password                                 %s\n", PASSWORD);
    printf("-D Database                                 %s\n", DATABASE);
                printf("-b Batch Size                                   %ld\n", (long)
BATCH);
                printf("-p TDS packet size                              %ld\n",
(long) DEFLDPACKSIZE);
                printf("-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. 4.41
//                                                      Copyright    Microsoft,
1996, 1997, 1998, 1999, 2000, 2001
//              Purpose:        Source file for database loader string functions


// Includes
#include "tpcc.h"
#include <string.h>
#include <ctype.h>

//=============================================
========================
//
// Function name: MakeAddress
//
//=============================================
========================

void MakeAddress(char *street_1,
                                                        char *street_2,
                                                        char *city,
                                                        char *state,
                                                        char *zip)
{

#ifdef DEBUG
    printf("[%ld]DBG:          Entering          MakeAddress()\n",          (int)
GetCurrentThreadId());
#endif

    MakeAlphaString (10, 20, ADDRESS_LEN, street_1);
    MakeAlphaString (10, 20, ADDRESS_LEN, street_2);
    MakeAlphaString (10, 20, ADDRESS_LEN, city);
    MakeAlphaString ( 2,  2, STATE_LEN, state);
    MakeZipNumberString( 9,  9, ZIP_LEN, zip);

#ifdef DEBUG
    printf("[%ld]DBG:  MakeAddress:  street_1:  %s,  street_2:  %s,  city:  %s,
state: %s, zip: %s\n",
                                                (int) GetCurrentThreadId(), street_1,
street_2, city, state, zip);
#endif

    return;

}


//=============================================
========================
//
// Function name: LastName
//
//=============================================
========================
```

```
void LastName(int  num,
                                char *name)
{
   static char *n[] =
   {
                        "BAR" , "OUGHT", "ABLE" , "PRI"  , "PRES",
                        "ESE" , "ANTI" , "CALLY", "ATION", "EING"
   };

#ifdef DEBUG
   printf("[%ld]DBG: Entering LastName()\n", (int) GetCurrentThreadId());
#endif

   if ((num >= 0) && (num < 1000))
   {
                        strcpy(name, n[(num/100)%10]);
                        strcat(name, n[(num/10)%10]);
                        strcat(name, n[(num/1)%10]);

                        if (strlen(name) < LAST_NAME_LEN)
                        {
                                PaddString(LAST_NAME_LEN,
name);
                        }

   }
   else
   {
                        printf("\nError in LastName()... num <%ld> out
of range (0,999)\n", num);
                        exit(-1);
   }


#ifdef DEBUG
   printf("[%ld]DBG: LastName: num = [%d] ==> [%d][%d][%d]\n",
                                        (int)  GetCurrentThreadId(),  num,
num/100, (num/10)%10, num%10);
            printf("[%ld]DBG:  LastName:  String  =  %s\n",  (int)
GetCurrentThreadId(), name);
#endif

   return;

}


//=============================================
==========================
//
// Function name: MakeAlphaString
//
//=============================================
==========================

//philipdu 08/13/96 Changed MakeAlphaString to use A-Z, a-z, and 0-9 in
//accordance with spec see below:
//The spec says:
//4.3.2.2       The notation random a-string [x .. y]
//(respectively, n-string [x .. y]) represents a string of random alphanumeric
//(respectively, numeric) characters of a random length of minimum x,
maximum y,
//and mean (y+x)/2. Alphanumerics are A..Z, a..z, and 0..9.  The only other
//requirement is that the character set used "must be able to represent a
minimum
//of 128 different characters".  We are using 8-bit chars, so this is a non
issue.
//It is completely unreasonable to stuff non-printing chars into the text fields.
//-CLevine 08/13/96

int MakeAlphaString( int  x, int  y, int  z, char *str)
{
                int                     len;
                int                     i;
   char    cc = 'a';
                static          char            chArray[]               =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
";
                static          int             chArrayMax = 61;

#ifdef DEBUG
   printf("[%ld]DBG:        Entering        MakeAlphaString()\n",        (int)
GetCurrentThreadId());
#endif

                len= RandomNumber(x, y);

                for (i=0; i<len; i++)
                {
                        cc = chArray[RandomNumber(0, chArrayMax)];
      str[i] = cc;
   }
                //if ( len < z )
                //              memset(str+len, ' ', z - len);
                str[len] = 0;

   return len;
}


//=============================================
==========================
//
// Function name: MakeOriginalAlphaString
//
//=============================================
==========================

int MakeOriginalAlphaString(int  x,

                int  y,

                        int  z,

                char *str,

                int  percent)
{
   int                  len;
   int                  val;
   int                  start;

#ifdef DEBUG
   printf("[%ld]DBG:      Entering      MakeOriginalAlphaString()\n",      (int)
GetCurrentThreadId());
#endif

   // verify prercentage is valid
   if ((percent < 0) || (percent > 100))
   {
                        printf("MakeOrigianlAlphaString:                Invalid
percentage: %d\n", percent);
                        exit(-1);
   }

   // verify string is at least 8 chars in length
   if ((x + y) <= 8)
   {
                        printf("MakeOriginalAlphaString:    string    length
must be >= 8\n");
                        exit(-1);
   }

   // Make Alpha String
   len = MakeAlphaString(x,y, z, str);

   val = RandomNumber(1,100);
   if (val <= percent)
   {
                        start = RandomNumber(0, len - 8);
                        strncpy(str + start, "ORIGINAL", 8);
   }


#ifdef DEBUG
   printf("[%ld]DBG: MakeOriginalAlphaString: : %s\n",
                                        (int) GetCurrentThreadId(), str);
#endif

   return strlen(str);
}


//=============================================
==========================
//
// Function name: MakeNumberString
//
//=============================================
==========================
int MakeNumberString(int  x, int  y, int  z, char *str)
{
        char tmp[16];

        //MakeNumberString is always called MakeZipNumberString(16,
16, 16, string)
```

```
		memset(str, '0', 16);
		itoa(RandomNumber(0, 99999999), tmp, 10);
		memcpy(str, tmp, strlen(tmp));

		itoa(RandomNumber(0, 99999999), tmp, 10);
		memcpy(str+8, tmp, strlen(tmp));

		str[16] = 0;

   return 16;
}


//=============================================
==========================
//
// Function name: MakeZipNumberString
//
//=============================================
==========================
int MakeZipNumberString(int  x, int  y, int  z, char *str)
{
		char tmp[16];

		//MakeZipNumberString	is	always	called
MakeZipNumberString(9, 9, 9, string)

		strcpy(str, "000011111");

		itoa(RandomNumber(0, 9999), tmp, 10);
		memcpy(str, tmp, strlen(tmp));

   return 9;
}


//=============================================
==========================
//
// Function name: InitString
//
//=============================================
==========================
void InitString(char *str, int len)
{
#ifdef DEBUG
   printf("[%ld]DBG: Entering InitString()\n", (int) GetCurrentThreadId());
#endif

		memset(str, ' ', len);
   str[len] = 0;
}


//=============================================
==========================
// Function name: InitAddress
//
// Description:
//
//=============================================
==========================

void InitAddress(char *street_1, char *street_2, char *city, char *state, char
*zip)
{
		memset(street_1, ' ', ADDRESS_LEN+1);
		memset(street_2, ' ', ADDRESS_LEN+1);
		memset(city, ' ', ADDRESS_LEN+1);

   street_1[ADDRESS_LEN+1] = 0;
   street_2[ADDRESS_LEN+1] = 0;
   city[ADDRESS_LEN+1]     = 0;

		memset(state, ' ', STATE_LEN+1);
   state[STATE_LEN+1] = 0;

		memset(zip, ' ', ZIP_LEN+1);
   zip[ZIP_LEN+1] = 0;
}


//=============================================
==========================
//
// Function name: PaddString
//
//=============================================
==========================

void PaddString(int max, char *name)
{
		int			len;

		len = strlen(name);
		if ( len < max )
			memset(name+len, ' ', max - len);
		name[max] = 0;

		return;
}
```

Appendix C Tunable Parameters

Disabled Windows Services

Alerter
Automatic Updates
Computer Browser
Cryptographic Services
DHCP Client
Distributed File System
Distributed Link Tracking Client
DNS Client
Error Reporting Service
HID Input Service
IPSEC Policy Agent
IMAPI CD-Burning COM Service
Kerberos Key Distribution Center
License Logging Service
Messenger
Microsoft Search
Print Spooler
Remote Registry Service
Removable Storage
Secondary Logon
System Event Notification
Task Scheduler
Telnet
Terminal Services Session Directory
WebClient
Windows Audio
Wireless Configuration

Server System Configuratiopn

System Information report written at: 12/10/02 10:00:45
System Name: TPC-RX5670
[System Summary]

Item Value
OS Name Microsoft® Windows® .NET Enterprise Server
Version 5.2.3663 Build 3663
OS Manufacturer Microsoft Corporation
System Name TPC-RX5670
System Manufacturer hp
System Model server rx5670
System Type Itanium (TM) -based System
Processor ia64 Family 31 Model 0 Stepping 6 GenuineIntel ~1000 Mhz

Processor ia64 Family 31 Model 0 Stepping 6 GenuineIntel ~1000 Mhz

Processor ia64 Family 31 Model 0 Stepping 6 GenuineIntel ~1000 Mhz

Processor ia64 Family 31 Model 0 Stepping 6 GenuineIntel ~1000 Mhz

BIOS Version/Date HP ia64_everest_release.view.02.00-0, 10/1/2002

SMBIOS Version 2.3
Windows Directory C:\WINDOWS
System Directory C:\WINDOWS\system32
Boot Device \Device\HarddiskVolume23
Locale United States
Hardware Abstraction Layer Version = "5.2.3663.0 (main.020715-1506)"
User Name TPC-RX5670\Administrator
Time Zone Pacific Standard Time
Total Physical Memory 49,152.00 MB
Available Physical Memory 46.81 GB
Total Virtual Memory 97.05 GB
Available Virtual Memory 95.54 GB
Page File Space 49.05 GB
Page File C:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

Resource Device
I/O Port 0x00002000-0x00003FFF PCI bus
I/O Port 0x00002000-0x00003FFF PCI standard PCI-to-PCI bridge

I/O Port 0x00002000-0x00003FFF LSI Logic 53C1010-66 Device

Generic Bus
PCI bus

Generic Bus
PCI bus

I/O Port 0x00006000-0x00007FFF PCI bus
I/O Port 0x00006000-0x00007FFF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x0000E000-0x0000FFFF PCI bus
I/O Port 0x0000E000-0x0000FFFF Smart Array 5300 Controller (Non-Miniport)

Memory Address 0x80000000-0xFDFFFFFF Generic Bus
Memory Address 0x80000000-0xFDFFFFFF PCI bus
Memory Address 0x80000000-0xFDFFFFFF LSI Logic 53C896 Device

Memory Address 0x90000000-0x9FFFFFFF PCI bus
Memory Address 0x90000000-0x9FFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0x90000000-0x9FFFFFFF Broadcom NetXtreme Gigabit Ethernet

Generic Bus
PCI bus

Generic Bus
PCI bus

Generic Bus
PCI bus

Generic Bus
PCI bus

Generic Bus
PCI bus

Generic Bus
PCI bus

I/O Port 0x0000C000-0x0000DFFF PCI bus
I/O Port 0x0000C000-0x0000DFFF Smart Array 5300 Controller (Non-Miniport)

I/O Port 0x00008000-0x00009FFF PCI bus
I/O Port 0x00008000-0x00009FFF Smart Array 5300 Controller (Non-Miniport)

[DMA]

Resource Device Status

[Forced Hardware]

Device PNP Device ID

[I/O]

Resource Device Status

0x00000000-0x00001FFF PCI bus OK
0x00000E00-0x00000EFF LSI Logic 53C896 Device OK
0x00000D00-0x00000DFF LSI Logic 53C896 Device OK
0x00002000-0x00003FFF PCI bus OK
0x00002000-0x00003FFF PCI standard PCI-to-PCI bridge OK

0x00002000-0x00003FFF LSI Logic 53C1010-66 Device OK

0x00004000-0x00005FFF PCI bus OK
0x00006000-0x00007FFF PCI bus OK
0x00006000-0x00007FFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0x00008000-0x00009FFF PCI bus OK
0x00008000-0x00009FFF Smart Array 5300 Controller (Non-Miniport) OK
0x00008100-0x000081FF Smart Array 5300 Controller (Non-Miniport) OK
0x0000A000-0x0000BFFF PCI bus OK
0x0000C000-0x0000DFFF PCI bus OK
0x0000C000-0x0000DFFF Smart Array 5300 Controller (Non-Miniport) OK
0x0000C100-0x0000C1FF Smart Array 5300 Controller (Non-Miniport) OK
0x0000E000-0x0000FFFF PCI bus OK
0x0000E000-0x0000FFFF Smart Array 5300 Controller (Non-Miniport) OK
0x0000E100-0x0000E1FF Smart Array 5300 Controller (Non-Miniport) OK

[IRQs]

Resource Device Status
IRQ 20 Microsoft ACPI-Compliant System OK
IRQ 17 LSI Logic 53C896 Device OK
IRQ 18 LSI Logic 53C896 Device OK
IRQ 29 LSI Logic 53C1010-66 Device OK
IRQ 27 Broadcom NetXtreme Gigabit Ethernet OK
IRQ 49 QLogic QLA23xx PCI Fibre Channel Adapter OK

IRQ 64 Smart Array 5300 Controller (Non-Miniport) OK

IRQ 60 Smart Array 5300 Controller (Non-Miniport) OK

IRQ 86 Smart Array 5300 Controller (Non-Miniport) OK

IRQ 82 Smart Array 5300 Controller (Non-Miniport) OK

IRQ 97 Smart Array 5300 Controller (Non-Miniport) OK

IRQ 93 Smart Array 5300 Controller (Non-Miniport) OK

[Memory]

Resource Device Status
0x80000000-0xFDFFFFFF Generic Bus OK
0x80000000-0xFDFFFFFF PCI bus OK
0x80000000-0xFDFFFFFF LSI Logic 53C896 Device OK
0xFF5B0000-0xFF5B0003 OK
0x80007000-0x8000700F PCI Simple Communications Controller OK

0x80006000-0x80006FFF PCI Serial Port OK
0x80005000-0x800053FF LSI Logic 53C896 Device OK
0x80002000-0x80003FFF LSI Logic 53C896 Device OK
0x80004000-0x800043FF LSI Logic 53C896 Device OK
0x90000000-0x9FFFFFFF PCI bus OK
0x90000000-0x9FFFFFFF PCI standard PCI-to-PCI bridge OK

0x90000000-0x9FFFFFFF Broadcom NetXtreme Gigabit Ethernet OK

0x90014000-0x900143FF LSI Logic 53C1010-66 Device OK

0x90010000-0x90011FFF LSI Logic 53C1010-66 Device OK

0xA0000000-0xAFFFFFFF PCI bus OK
0xB0000000-0xBFFFFFFF PCI bus OK
0xB0020000-0xB0020FFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0xC0000000-0xCFFFFFFF PCI bus OK
0xC0440000-0xC047FFFF Smart Array 5300 Controller (Non-Miniport) OK
0xC0300000-0xC03FFFFF Smart Array 5300 Controller (Non-Miniport) OK
0xC0400000-0xC043FFFF Smart Array 5300 Controller (Non-Miniport) OK
0xC0100000-0xC01FFFFF Smart Array 5300 Controller (Non-Miniport) OK
0xD0000000-0xDFFFFFFF PCI bus OK
0xE0000000-0xEFFFFFFF PCI bus OK
0xE0440000-0xE047FFFF Smart Array 5300 Controller (Non-Miniport) OK
0xE0300000-0xE03FFFFF Smart Array 5300 Controller (Non-Miniport) OK
0xE0400000-0xE043FFFF Smart Array 5300 Controller (Non-Miniport) OK
0xE0100000-0xE01FFFFF Smart Array 5300 Controller (Non-Miniport) OK
0xF0000000-0xFDFFFFFF PCI bus OK
0xF0440000-0xF047FFFF Smart Array 5300 Controller (Non-Miniport) OK
0xF0300000-0xF03FFFFF Smart Array 5300 Controller (Non-Miniport) OK
0xF0400000-0xF043FFFF Smart Array 5300 Controller (Non-Miniport) OK
0xF0100000-0xF01FFFFF Smart Array 5300 Controller (Non-Miniport) OK

[Components]

[Multimedia]

[Audio Codecs]

CODEC Manufacturer Description Status File Version
Size Creation Date
c:\windows\system32\msgsm32.acm Microsoft Corporation
OK C:\WINDOWS\system32\MSGSM32.ACM
5.2.3663.0 (main.020715-1506) 66.50 KB (68,096 bytes) 7/22/2002 5:00 AM
c:\windows\system32\imaadp32.acm Microsoft Corporation
OK C:\WINDOWS\system32\IMAADP32.ACM
5.2.3663.0 (main.020715-1506) 55.00 KB (56,320 bytes) 7/22/2002 5:00 AM
c:\windows\system32\tssoft32.acm DSP GROUP, INC.
OK C:\WINDOWS\system32\TSSOFT32.ACM 1.01
29.00 KB (29,696 bytes) 7/22/2002 5:00 AM
c:\windows\system32\msadp32.acm Microsoft Corporation
OK C:\WINDOWS\system32\MSADP32.ACM
5.2.3663.0 (main.020715-1506) 48.50 KB (49,664 bytes) 7/22/2002 5:00 AM
c:\windows\system32\msg711.acm Microsoft Corporation
OK C:\WINDOWS\system32\MSG711.ACM
5.2.3663.0 (main.020715-1506) 33.00 KB (33,792 bytes) 7/22/2002 5:00 AM

[Video Codecs]

CODEC Manufacturer Description Status File Version
Size Creation Date
c:\windows\system32\msvidc32.dll Microsoft Corporation
OK C:\WINDOWS\system32\MSVIDC32.DLL
5.2.3663.0 (main.020715-1506) 67.00 KB (68,608 bytes) 7/22/2002 5:00 AM
c:\windows\system32\msrle32.dll Microsoft Corporation
OK C:\WINDOWS\system32\MSRLE32.DLL
5.2.3663.0 (main.020715-1506) 24.50 KB (25,088 bytes) 7/22/2002 5:00 AM

[CD-ROM]

Item Value
Drive D:
Description CD-ROM Drive
Media Loaded No

Media Type CD-ROM
Name HP DVD-ROM 305 SCSI CdRom Device
Manufacturer (Standard CD-ROM drives)
Status OK
Transfer Rate Not Available
SCSI Target ID 2
PNP Device ID SCSI\CDROM&VEN_HP&PROD_DVD-ROM_305&REV_1.01\5&968E4F&0&020
Driver c:\windows\system32\drivers\cdrom.sys (5.2.3663.0 (main.020715-1506), 143.50 KB (146,944 bytes), 7/22/2002 5:00 AM)

[Sound Device]

Item Value

[Display]

Item Value

[Infrared]

Item Value

[Input]

[Keyboard]

Item Value

[Pointing Device]

Item Value

[Modem]

Item Value

[Network]

[Adapter]

Item Value
Name [00000001] Broadcom NetXtreme Gigabit Ethernet
Adapter Type Ethernet 802.3
Product Type Broadcom NetXtreme Gigabit Ethernet
Installed Yes
PNP Device ID PCI\VEN_14E4&DEV_1645&SUBSYS_1300103C&REV_15\5&2B7E0477&0&28F0
Last Reset 12/10/2002 9:56 AM
Index 1
Service Name b57nd
IP Address 15.1.100.26
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled Yes
DHCP Server 15.1.100.1
DHCP Lease Expires 12/18/2002 9:57 AM
DHCP Lease Obtained 12/10/2002 9:57 AM
MAC Address 00:30:6E:05:A9:7D
Memory Address 0x90000000-0x9FFFFFFF
IRQ Channel IRQ 27
Driver c:\windows\system32\drivers\b57xp64.sys (2.67.0.0 built by: WinDDK, 475.25 KB (486,656 bytes), 8/23/2002 6:04 AM)

Name [00000002] RAS Async Adapter
Adapter Type Not Available
Product Type RAS Async Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 12/10/2002 9:56 AM
Index 2
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000003] WAN Miniport (L2TP)
Adapter Type Not Available
Product Type WAN Miniport (L2TP)
Installed Yes
PNP Device ID ROOT\MS_L2TPMINIPORT\0000
Last Reset 12/10/2002 9:56 AM
Index 3
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\rasl2tp.sys (5.2.3663.0 (main.020715-1506), 181.38 KB (185,728 bytes), 7/22/2002 5:00 AM)

Name [00000004] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPTP)
Installed Yes
PNP Device ID ROOT\MS_PPTPMINIPORT\0000
Last Reset 12/10/2002 9:56 AM
Index 4
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30
Driver c:\windows\system32\drivers\raspptp.sys (5.2.3663.0 (main.020715-1506), 176.75 KB (180,992 bytes), 7/22/2002 5:00 AM)

Name [00000005] WAN Miniport (PPPOE)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPPOE)
Installed Yes
PNP Device ID ROOT\MS_PPPOEMINIPORT\0000
Last Reset 12/10/2002 9:56 AM
Index 5
Service Name RasPppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30
Driver c:\windows\system32\drivers\raspppoe.sys (5.2.3663.0 (main.020715-1506), 115.13 KB (117,888 bytes), 7/22/2002 5:00 AM)

Name [00000006] Direct Parallel
Adapter Type Not Available
Product Type Direct Parallel
Installed Yes
PNP Device ID ROOT\MS_PTIMINIPORT\0000
Last Reset 12/10/2002 9:56 AM
Index 6
Service Name Raspti
IP Address Not Available
IP Subnet Not Available

Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\raspti.sys (5.2.3663.0 (main.020715-1506), 47.25 KB (48,384 bytes), 7/22/2002 5:00 AM)

Name [00000007] WAN Miniport (IP)
Adapter Type Not Available
Product Type WAN Miniport (IP)
Installed Yes
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 12/10/2002 9:56 AM
Index 7
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\ndiswan.sys (5.2.3663.0 (main.020715-1506), 260.25 KB (266,496 bytes), 7/22/2002 5:00 AM)

Name [00000008] Intel(R) PRO/1000 T Server Adapter
Adapter Type Not Available
Product Type Intel(R) PRO/1000 T Server Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 12/10/2002 9:56 AM
Index 8
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000009] Myricom Myrinet Adapter
Adapter Type Not Available
Product Type Myricom Myrinet Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 12/10/2002 9:56 AM
Service Name GM
Index 9
IP Address 15.1.100.26
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled Yes
DHCP Server 255.255.255.255
DHCP Lease Expires 12/31/1969 3:59 PM
DHCP Lease Obtained 10/30/2002 4:34 PM
MAC Address 00:30:6E:05:A9:7D

[Protocol]

Item Value
Name MSAFD Tcpip [TCP/IP]
Connectionless Service No
Guarantees Delivery Yes
Guarantees Sequencing Yes
Maximum Address Size 16 bytes
Maximum Message Size 0 bytes
Message Oriented No
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data Yes
Supports Graceful Closing Yes
Supports Guaranteed Bandwidth No
Supports Multicasting No

Name MSAFD Tcpip [UDP/IP]
Connectionless Service Yes
Guarantees Delivery No
Guarantees Sequencing No
Maximum Address Size 16 bytes
Maximum Message Size 63.93 KB (65,467 bytes)
Message Oriented Yes
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting Yes

Name RSVP UDP Service Provider
Connectionless Service Yes
Guarantees Delivery No
Guarantees Sequencing No
Maximum Address Size 16 bytes
Maximum Message Size 63.93 KB (65,467 bytes)
Message Oriented Yes
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption Yes
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting Yes

Name RSVP TCP Service Provider
Connectionless Service No
Guarantees Delivery Yes
Guarantees Sequencing Yes
Maximum Address Size 16 bytes
Maximum Message Size 0 bytes
Message Oriented No
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption Yes
Supports Expedited Data Yes
Supports Graceful Closing Yes
Supports Guaranteed Bandwidth No
Supports Multicasting No

[WinSock]

Item Value
File c:\windows\system32\wsock32.dll
Size 23.00 KB (23,552 bytes)
Version 5.2.3663.0 (main.020715-1506)

[Ports]

[Serial]

Item Value

[Parallel]

Item Value

[Storage]

[Drives]

Item	Value
Drive	C:
Description	Local Fixed Disk
Compressed	No
File System	NTFS
Size	33.45 GB (35,919,794,176 bytes)
Free Space	26.22 GB (28,157,030,400 bytes)
Volume Name	
Volume Serial Number	D080A1C3

Drive	D:
Description	CD-ROM Disc

Drive	L:
Description	Local Fixed Disk
Compressed	Not Available
File System	Not Available
Size	Not Available
Free Space	Not Available
Volume Name	Not Available
Volume Serial Number	Not Available

Drive	Y:
Description	Network Connection
Provider Name	\\hptpc\c$

Drive	Z:
Description	Network Connection
Provider Name	\\ssdllab\d$

[Disks]

Item	Value
Description	\\.\PHYSICALDRIVE0
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	No
Media Type	Fixed hard disk
Partitions	Not Available
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	237.42 GB (254,926,103,040 bytes)
Total Cylinders	30,993
Total Sectors	497,902,545
Total Tracks	7,903,215
Tracks/Cylinder	255
Partition	Disk #0, Partition #0
Partition Size	237.30 GB (254,796,168,704 bytes)
Partition Starting Offset	136,314,880 bytes

Description	\\.\PHYSICALDRIVE2
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	No
Media Type	Fixed hard disk
Partitions	Not Available
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	949.65 GB (1,019,679,736,320 bytes)
Total Cylinders	123,969
Total Sectors	1,991,561,985
Total Tracks	31,612,095
Tracks/Cylinder	255
Partition	Disk #2, Partition #0
Partition Size	79.59 GB (85,458,944,000 bytes)
Partition Starting Offset	136,314,880 bytes
Partition	Disk #2, Partition #1
Partition Size	41.02 GB (44,040,192,000 bytes)
Partition Starting Offset	85,595,258,880 bytes
Partition	Disk #2, Partition #2
Partition Size	828.92 GB (890,043,891,712 bytes)
Partition Starting Offset	129,635,450,880 bytes

Description	\\.\PHYSICALDRIVE4
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	No
Media Type	Fixed hard disk
Partitions	Not Available
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	949.65 GB (1,019,679,736,320 bytes)
Total Cylinders	123,969
Total Sectors	1,991,561,985
Total Tracks	31,612,095
Tracks/Cylinder	255
Partition	Disk #4, Partition #0
Partition Size	79.59 GB (85,458,944,000 bytes)
Partition Starting Offset	136,314,880 bytes
Partition	Disk #4, Partition #1
Partition Size	41.02 GB (44,040,192,000 bytes)
Partition Starting Offset	85,595,258,880 bytes
Partition	Disk #4, Partition #2
Partition Size	828.92 GB (890,043,891,712 bytes)
Partition Starting Offset	129,635,450,880 bytes

Description	\\.\PHYSICALDRIVE3
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	No
Media Type	Fixed hard disk
Partitions	Not Available
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	949.65 GB (1,019,679,736,320 bytes)
Total Cylinders	123,969
Total Sectors	1,991,561,985
Total Tracks	31,612,095
Tracks/Cylinder	255
Partition	Disk #3, Partition #0
Partition Size	79.59 GB (85,458,944,000 bytes)
Partition Starting Offset	136,314,880 bytes
Partition	Disk #3, Partition #1
Partition Size	41.02 GB (44,040,192,000 bytes)
Partition Starting Offset	85,595,258,880 bytes
Partition	Disk #3, Partition #2
Partition Size	828.92 GB (890,043,891,712 bytes)
Partition Starting Offset	129,635,450,880 bytes

Description	\\.\PHYSICALDRIVE1
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	No
Media Type	Fixed hard disk
Partitions	Not Available
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	949.65 GB (1,019,679,736,320 bytes)
Total Cylinders	123,969
Total Sectors	1,991,561,985
Total Tracks	31,612,095
Tracks/Cylinder	255
Partition	Disk #1, Partition #0
Partition Size	79.59 GB (85,458,944,000 bytes)
Partition Starting Offset	136,314,880 bytes
Partition	Disk #1, Partition #1
Partition Size	41.02 GB (44,040,192,000 bytes)

Partition Starting Offset 85,595,258,880 bytes
Partition Disk #1, Partition #2
Partition Size 828.92 GB (890,043,891,712 bytes)
Partition Starting Offset 129,635,450,880 bytes

Description \\.\PHYSICALDRIVE5
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 949.65 GB (1,019,679,736,320 bytes)
Total Cylinders 123,969
Total Sectors 1,991,561,985
Total Tracks 31,612,095
Tracks/Cylinder 255
Partition Disk #5, Partition #0
Partition Size 79.59 GB (85,458,944,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #5, Partition #1
Partition Size 41.02 GB (44,040,192,000 bytes)
Partition Starting Offset 85,595,258,880 bytes
Partition Disk #5, Partition #2
Partition Size 828.92 GB (890,043,891,712 bytes)
Partition Starting Offset 129,635,450,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model FUJITSU MAJ3364MC SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 1
SCSI Target ID 0
Sectors/Track 63
Size 33.91 GB (36,413,314,560 bytes)
Total Cylinders 4,427
Total Sectors 71,119,755
Total Tracks 1,128,885
Tracks/Cylinder 255
Partition Disk #7, Partition #0
Partition Size 345.12 MB (361,880,064 bytes)
Partition Starting Offset 32,256 bytes

Partition Disk #7, Partition #1
Partition Size 33.45 GB (35,919,797,760 bytes)
Partition Starting Offset 493,516,800 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model QLOGIC PSEUDO DEVICE SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 0
SCSI Target ID 127
Sectors/Track 0
Size 0 bytes
Total Cylinders 0
Total Sectors 0
Total Tracks 0
Tracks/Cylinder 0

[SCSI]

Item Value
Name LSI Logic 53C896 Device
Manufacturer LSI Logic Inc.
Status OK
PNP Device ID
PCI\VEN_1000&DEV_000B&SUBSYS_00000000&REV_07\4&4F5EBC7&0&10
I/O Port 0x00000E00-0x00000EFF
Memory Address 0x80005000-0x800053FF
Memory Address 0x80002000-0x80003FFF
IRQ Channel IRQ 17
Driver c:\windows\system32\drivers\sym_hi.sys (5.1.3563.0 (lab01_n(storbuild).011003-2142), 61.75 KB (63,232 bytes), 7/22/2002 5:00 AM)

Name LSI Logic 53C896 Device
Manufacturer LSI Logic Inc.
Status OK
PNP Device ID
PCI\VEN_1000&DEV_000B&SUBSYS_00000000&REV_07\4&4F5EBC7&0&11
I/O Port 0x00000D00-0x00000DFF
Memory Address 0x80004000-0x800043FF
Memory Address 0x80000000-0xFDFFFFFF
IRQ Channel IRQ 18
Driver c:\windows\system32\drivers\sym_hi.sys (5.1.3563.0 (lab01_n(storbuild).011003-2142), 61.75 KB (63,232 bytes), 7/22/2002 5:00 AM)

Name LSI Logic 53C1010-66 Device
Status Error
Manufacturer LSI Logic Inc.
PNP Device ID
PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&2B7E6A47&0&0808
Driver c:\windows\system32\drivers\sym_u3.sys (5.1.3563.0 (lab01_n(storbuild).011003-2142), 67.75 KB (69,376 bytes), 7/22/2002 5:00 AM)

Name LSI Logic 53C1010-66 Device
Manufacturer LSI Logic Inc.
Status OK
PNP Device ID
PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&2B7E6A47&0&0908
I/O Port 0x00002000-0x00003FFF
Memory Address 0x90014000-0x900143FF
Memory Address 0x90010000-0x90011FFF
IRQ Channel IRQ 29
Driver c:\windows\system32\drivers\sym_u3.sys (5.1.3563.0 (lab01_n(storbuild).011003-2142), 67.75 KB (69,376 bytes), 7/22/2002 5:00 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&19EBB955&0&08
I/O Port 0x00006000-0x00007FFF
Memory Address 0xB0020000-0xB0020FFF
IRQ Channel IRQ 49
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 3 (W64 VI), 673.13 KB (689,280 bytes), 11/12/2002 9:19 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\4&15291AB&0&08
Memory Address 0xC0440000-0xC047FFFF
Memory Address 0xC0300000-0xC03FFFFF
I/O Port 0x00008100-0x000081FF
IRQ Channel IRQ 64
Driver c:\windows\system32\drivers\hpqcissb.sys (5.5.58.64 built by: WinDDK, 105.63 KB (108,160 bytes), 11/12/2002 2:56 PM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\4&15291AB&0&10

Memory Address 0xC0400000-0xC043FFFF
Memory Address 0xC0100000-0xC01FFFFF
I/O Port 0x00008000-0x00009FFF
IRQ Channel IRQ 60
Driver c:\windows\system32\drivers\hpqcissb.sys (5.5.58.64 built by: WinDDK, 105.63 KB (108,160 bytes), 11/12/2002 2:56 PM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\4&5D9CB86&0&08
Memory Address 0xE0440000-0xE047FFFF
Memory Address 0xE0300000-0xE03FFFFF
I/O Port 0x0000C100-0x0000C1FF
IRQ Channel IRQ 86
Driver c:\windows\system32\drivers\hpqcissb.sys (5.5.58.64 built by: WinDDK, 105.63 KB (108,160 bytes), 11/12/2002 2:56 PM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\4&5D9CB86&0&10
Memory Address 0xE0400000-0xE043FFFF
Memory Address 0xE0100000-0xE01FFFFF
I/O Port 0x0000C000-0x0000DFFF
IRQ Channel IRQ 82
Driver c:\windows\system32\drivers\hpqcissb.sys (5.5.58.64 built by: WinDDK, 105.63 KB (108,160 bytes), 11/12/2002 2:56 PM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\4&1E72F330&0&08
Memory Address 0xF0440000-0xF047FFFF
Memory Address 0xF0300000-0xF03FFFFF
I/O Port 0x0000E100-0x0000E1FF
IRQ Channel IRQ 97
Driver c:\windows\system32\drivers\hpqcissb.sys (5.5.58.64 built by: WinDDK, 105.63 KB (108,160 bytes), 11/12/2002 2:56 PM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\4&1E72F330&0&10
Memory Address 0xF0400000-0xF043FFFF
Memory Address 0xF0100000-0xF015FFFF
I/O Port 0x0000E000-0x0000FFFF
IRQ Channel IRQ 93
Driver c:\windows\system32\drivers\hpqcissb.sys (5.5.58.64 built by: WinDDK, 105.63 KB (108,160 bytes), 11/12/2002 2:56 PM)

[IDE]

Item Value

[Printing]

Name Driver Port Name Server Name

[Problem Devices]

Device PNP Device ID Error Code
Not Available ACPI\IPI0001\0 The drivers for this device are not installed.
PCI Simple Communications Controller
PCI\VEN_103C&DEV_1290&SUBSYS_1291103C&REV_01\4&4F5EBC7&0&08 The drivers for this device are not installed.
PCI Serial Port
PCI\VEN_103C&DEV_1048&SUBSYS_1282103C&REV_03\4&4F5EBC7&0&09 The drivers for this device are not installed.
LSI Logic 53C1010-66 Device
PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&2B7E6A47&0&0808 This device is disabled.

[USB]

Device PNP Device ID

[Software Environment]

[System Drivers]

Name	Description	File	Type	Started	Start Mode	State	Status	Error Control	Accept Pause	Accept Stop
abiosdsk	Abiosdsk	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Ignore	No	No
acpi	Microsoft ACPI Driver	c:\windows\system32\drivers\acpi.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No	Yes
acpiec	ACPIEC	c:\windows\system32\drivers\acpiec.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
adpu160m	adpu160m	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
adpu320	adpu320	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
afcnt	afcnt	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
afd	AFD Networking Support Environment	c:\windows\system32\drivers\afd.sys	Kernel Driver	Yes	Auto	Running	OK	Normal	No	Yes
aic78u2	aic78u2	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
aic78xx	aic78xx	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
aliide	AliIde	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
asyncmac	RAS Asynchronous Media Driver	c:\windows\system32\drivers\asyncmac.sys	Kernel Driver	No	Manual	Stopped	OK	Normal	No	No
atapi	atapi	c:\windows\system32\drivers\atapi.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
atdisk	Atdisk	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Ignore	No	No
ati2mtag	ati2mtag	c:\windows\system32\drivers\ati2mtag.sys	Kernel Driver	No	Manual	Stopped	OK	Ignore	No	No
atmarpc	ATM ARP Client Protocol	c:\windows\system32\drivers\atmarpc.sys	Kernel Driver	No	Manual	Stopped	OK	Normal	No	No
audstub	Audio Stub Driver	c:\windows\system32\drivers\audstub.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No	Yes
b57nd	Broadcom NetXtreme Gigabit Ethernet	c:\windows\system32\drivers\b57xp64.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No	Yes
beep	Beep	c:\windows\system32\drivers\beep.sys	Kernel Driver	Yes	System	Running	OK	Normal	No	Yes
cbidf2k	cbidf2k	c:\windows\system32\drivers\cbidf2k.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No

```
cdfs            Cdfs            c:\windows\system32\drivers\cdfs.sys    File
System Driver                   Yes             Disabled        Running OK
                Normal          No              Yes
cdrom           CD-ROM Driver
                c:\windows\system32\drivers\cdrom.sys   Kernel Driver Yes
                System          Running         OK              Normal  No
                Yes
changer         Changer         Not Available   Kernel Driver   No      System
                Stopped         OK              Ignore          No      No

clusdisk        Cluster Disk Driver
                c:\windows\system32\drivers\clusdisk.sys                Kernel
Driver          No              Disabled        Stopped         OK      Normal
                No              No
cmdide          CmdIde          Not Available   Kernel Driver   No      Disabled
                Stopped         OK              Normal          No      No

cpqarry2        cpqarry2        Not Available   Kernel Driver   No      Disabled
                Stopped         OK              Normal          No      No

cpqcisse        CPQCISSE        c:\windows\system32\drivers\cpqcisse.sys
                Kernel Driver   No              System          Stopped OK
                Normal          No              No
cpqcissm        cpqcissm        c:\windows\system32\drivers\cpqcissm.sys
                Kernel Driver   Yes             Boot            Running OK
                Normal          No              Yes
cpqfcalm        cpqfcalm        Not Available   Kernel Driver   No      Disabled
                Stopped         OK              Normal          No      No

crcdisk         CRC Disk Filter Driver
                c:\windows\system32\drivers\crcdisk.sys                 Kernel
Driver          Yes             Boot            Running         OK      Normal
                No              Yes
dac2w2k         dac2w2k         c:\windows\system32\drivers\dac2w2k.sys
                Kernel Driver   Yes             Boot            Running OK
                Normal          No              Yes
dfsdriver       DfsDriver       c:\windows\system32\drivers\dfs.sys     File
System Driver                   Yes             Boot            Running OK
                Normal          No              Yes
disk            Disk Driver     c:\windows\system32\drivers\disk.sys    Kernel
Driver          Yes             Boot            Running         OK      Normal
                No              Yes
dmboot          dmboot          c:\windows\system32\drivers\dmboot.sys
                Kernel Driver   No              Disabled        Stopped OK
                Normal          No              No
dmio            Logical Disk Manager Driver
                c:\windows\system32\drivers\dmio.sys   Kernel Driver Yes
                Boot            Running         OK              Normal  No
                Yes
dmload          dmload          c:\windows\system32\drivers\dmload.sys
                Kernel Driver   Yes             Boot            Running OK
                Normal          No              Yes

dpti2o          dpti2o          Not Available   Kernel Driver   No      Disabled
                Stopped         OK              Normal          No      No
e1000           Intel(R) PRO/1000 Device Driver
                c:\windows\system32\drivers\e1000645.sys                Kernel
Driver          No              Manual          Stopped         OK      Normal
                No              No
em              em              \??\c:\windows\system32\drivers\em.sys
                Kernel Driver   No              Manual          Stopped OK
                Normal          No              No
fastfat         Fastfat         c:\windows\system32\drivers\fastfat.sys File
System Driver                   No              Disabled        Stopped OK
                Normal          No              No
fdc             Fdc             c:\windows\system32\drivers\fdc.sys     Kernel
Driver          No              System          Stopped         OK      Ignore
                No              No
fips            Fips            c:\windows\system32\drivers\fips.sys    Kernel
Driver          Yes             System          Running         OK      Normal
                No              Yes
flpydisk        Flpydisk        c:\windows\system32\drivers\flpydisk.sys
                Kernel Driver   No              System          Stopped OK
                Ignore          No              No
ftdisk          Volume Manager Driver
                c:\windows\system32\drivers\ftdisk.sys  Kernel Driver Yes
                Boot            Running         OK              Normal  No
                Yes
gpc             Generic Packet Classifier
                c:\windows\system32\drivers\msgpc.sys   Kernel Driver Yes
                Manual          Running         OK              Normal  No
                Yes
hidusb          Microsoft HID Class Driver
                c:\windows\system32\drivers\hidusb.sys  Kernel Driver No
                Manual          Stopped         OK              Ignore  No
                No
hpn             hpn             c:\windows\system32\drivers\hpn.sys     Kernel
Driver          Yes             Boot            Running         OK      Normal
                No              Yes
hpqcissb        Smart Array Controllers Non-Miniport Bus Driver
                c:\windows\system32\drivers\hpqcissb.sys                Kernel
Driver          Yes             Boot            Running         OK      Normal
                No              Yes
hpqcissd        Smart Array Controllers Non-Miniport Disk Driver
                c:\windows\system32\drivers\hpqcissd.sys                Kernel
Driver          Yes             Boot            Running         OK      Normal
                No              Yes
http            HTTP            c:\windows\system32\drivers\http.sys    Kernel
Driver          No              Manual          Stopped         OK      Normal
                No              No
i2omgmt         i2omgmt         Not Available   Kernel Driver   No      System
                Stopped         OK              Normal          No      No

imapi           CD-Burning Filter Driver
                c:\windows\system32\drivers\imapi.sys   Kernel Driver No
                System          Stopped         OK              Normal  No
intelide        IntelIde        Not Available   Kernel Driver   No      Disabled
                Stopped         OK              Normal          No      No

ipfilterdriver  IP Traffic Filter Driver
                c:\windows\system32\drivers\ipfltdrv.sys                Kernel
Driver          No              Manual          Stopped         OK      Normal
                No              No
ipinip          IP in IP Tunnel Driver
                c:\windows\system32\drivers\ipinip.sys  Kernel Driver No
                Manual          Stopped         OK              Normal  No
                No
ipnat           IP Network Address Translator
                c:\windows\system32\drivers\ipnat.sys   Kernel Driver No
                Manual          Stopped         OK              Normal  No
                No
ipsec           IPSEC driver    c:\windows\system32\drivers\ipsec.sys   Kernel
Driver          Yes             System          Running         OK      Normal
                No              Yes
isapnp          isapnp          Not Available   Kernel Driver   No      Disabled
                Stopped         OK              Normal          No      No

kbdclass        Keyboard Class Driver
                c:\windows\system32\drivers\kbdclass.sys                Kernel
Driver          Yes             System          Running         OK      Normal
                No              Yes
kbdhid          Keyboard HID Driver
                c:\windows\system32\drivers\kbdhid.sys  Kernel Driver No
                System          Stopped         OK              Ignore  No
                No
ksecdd          KSecDD          c:\windows\system32\drivers\ksecdd.sys
                Kernel Driver   Yes             Boot            Running OK
                Normal          No              Yes
lp6nds35        lp6nds35        Not Available   Kernel Driver   No      Disabled
                Stopped         OK              Normal          No      No

mnmdd           mnmdd           Not Available   Kernel Driver   No      System
                Stopped         OK              Ignore          No      No

modem           Modem           c:\windows\system32\drivers\modem.sys
                Kernel Driver   No              Manual          Stopped OK
                Ignore          No              No
mouclass        Mouse Class Driver
                c:\windows\system32\drivers\mouclass.sys                Kernel
Driver          Yes             System          Running         OK      Normal
                No              Yes
mouhid          Mouse HID Driver
                c:\windows\system32\drivers\mouhid.sys                  Kernel
Driver          No              Manual          Stopped         OK      Ignore
                No              No
```

```
mountmgr  Mount Point Manager
          c:\windows\system32\drivers\mountmgr.sys          Kernel
Driver    Yes       Boot      Running   OK        Normal
          No        Yes
mraid35x  mraid35x  Not Available Kernel Driver No        Disabled
          Stopped   OK        Normal    No        No

mrxdav    WebDav Client Redirector
          c:\windows\system32\drivers\mrxdav.sys            File
System Driver       No        Manual    Stopped   OK
          Normal    No        No
mrxsmb    MRXSMB    c:\windows\system32\drivers\mrxsmb.sys
          File System Driver  Yes       System    Running
          OK        Normal    No        Yes
msfs      Msfs      c:\windows\system32\drivers\msfs.sys    File
System Driver       Yes       System    Running   OK
          Normal    No        Yes
mup       Mup       c:\windows\system32\drivers\mup.sys     File
System Driver       Yes       Boot      Running   OK
          Normal    No        Yes
myrinetadapter      Myricom Myrinet Driver
          c:\windows\system32\drivers\gm.sys    Kernel Driver No
          Manual    Stopped   OK        Normal    No
          No
ndis      NDIS System Driver
          c:\windows\system32\drivers\ndis.sys  Kernel Driver Yes
          Boot      Running   OK        Normal    No
          Yes
ndistapi  Remote Access NDIS TAPI Driver
          c:\windows\system32\drivers\ndistapi.sys          Kernel
Driver    Yes       Manual    Running   OK        Normal
          No        Yes
ndisuio   NDIS Usermode I/O Protocol
          c:\windows\system32\drivers\ndisuio.sys           Kernel
Driver    No        Manual    Stopped   OK        Normal
          No        No
ndiswan   Remote Access NDIS WAN Driver
          c:\windows\system32\drivers\ndiswan.sys           Kernel
Driver    Yes       Manual    Running   OK        Normal
          No        Yes
ndproxy   NDIS Proxy  c:\windows\system32\drivers\ndproxy.sys
          Kernel Driver Yes   Manual    Running   OK
          Normal    No        Yes
netbios   NetBIOS Interface
          c:\windows\system32\drivers\netbios.sys           File
System Driver       Yes       System    Running   OK
          Normal    No        Yes
netbt     NetBios over Tcpip
          c:\windows\system32\drivers\netbt.sys Kernel Driver Yes
          System    Running   OK        Normal    No
          Yes
```

```
nfrd960   nfrd960   Not Available Kernel Driver No        Disabled
          Stopped   OK        Normal    No        No
npfs      Npfs      c:\windows\system32\drivers\npfs.sys    File
System Driver       Yes       System    Running   OK
          Normal    No        Yes
ntfs      Ntfs      c:\windows\system32\drivers\ntfs.sys    File
System Driver       Yes       Disabled  Running   OK
          Normal    No        Yes
null      Null      c:\windows\system32\drivers\null.sys    Kernel
Driver    Yes       System    Running   OK        Normal
          No        Yes
partmgr   Partition Manager
          c:\windows\system32\drivers\partmgr.sys           Kernel
Driver    Yes       Boot      Running   OK        Normal
          No        Yes
pci       PCI Bus Driver      c:\windows\system32\drivers\pci.sys
          Kernel Driver Yes   Boot      Running   OK
          Critical  No        Yes
pciide    PCIIde    Not Available Kernel Driver No        Disabled
          Stopped   OK        Normal    No        No

pcmcia    Pcmcia    c:\windows\system32\drivers\pcmcia.sys
          Kernel Driver No    Disabled  Stopped   OK
          Normal    No        No
pdcomp    PDCOMP    Not Available Kernel Driver No        Manual
          Stopped   OK        Ignore    No        No

pdframe   PDFRAME   Not Available Kernel Driver No        Manual
          Stopped   OK        Ignore    No        No

pdreli    PDRELI    Not Available Kernel Driver No        Manual
          Stopped   OK        Ignore    No        No

pdrframe  PDRFRAME  Not Available Kernel Driver No        Manual
          Stopped   OK        Ignore    No        No

pptpminiport WAN Miniport (PPTP)
          c:\windows\system32\drivers\raspptp.sys           Kernel
Driver    Yes       Manual    Running   OK        Normal
          No        Yes
processor Processor Driver
          c:\windows\system32\drivers\processr.sys          Kernel
Driver    Yes       Manual    Running   OK        Normal
          No        Yes
ptilink   Direct Parallel Link Driver
          c:\windows\system32\drivers\ptilink.sys Kernel Driver Yes
          Manual    Running   OK        Normal    No
          Yes
ql1080    ql1080    Not Available Kernel Driver No        Disabled
          Stopped   OK        Normal    No        No
```

```
ql10wnt   Ql10wnt   Not Available Kernel Driver No        Disabled
          Stopped   OK        Normal    No        No
ql12160   ql12160   Not Available Kernel Driver No        Disabled
          Stopped   OK        Normal    No        No

ql1240    ql1240    Not Available Kernel Driver No        Disabled
          Stopped   OK        Normal    No        No

ql1280    ql1280    Not Available Kernel Driver No        Disabled
          Stopped   OK        Normal    No        No

ql2100    ql2100    Not Available Kernel Driver No        Disabled
          Stopped   OK        Normal    No        No

ql2200    ql2200    Not Available Kernel Driver No        Disabled
          Stopped   OK        Normal    No        No

ql2300    ql2300    c:\windows\system32\drivers\ql2300.sysKernel
Driver    Yes       Boot      Running   OK        Normal
          No        Yes
qlvika    qlvika    c:\windows\system32\drivers\qlvika.sys Kernel
Driver    Yes       Auto      Running   OK        Normal
          No        Yes
rasacd    Remote Access Auto Connection Driver
          c:\windows\system32\drivers\rasacd.sys Kernel Driver Yes
          System    Running   OK        Normal    No
          Yes
rasl2tp   WAN Miniport (L2TP)
          c:\windows\system32\drivers\rasl2tp.sysKernel Driver Yes
          Manual    Running   OK        Normal    No
          Yes
raspppoe  Remote Access PPPOE Driver
          c:\windows\system32\drivers\raspppoe.sys          Kernel
Driver    Yes       Manual    Running   OK        Normal
          No        Yes
raspti    Direct Parallel
          c:\windows\system32\drivers\raspti.sys Kernel Driver Yes
          Manual    Running   OK        Normal    No
          Yes
rdbss     Rdbss     c:\windows\system32\drivers\rdbss.sys   File
System Driver       Yes       System    Running   OK
          Normal    No        Yes
rdpcdd    RDPCDD    c:\windows\system32\drivers\rdpcdd.sys
          Kernel Driver Yes   System    Running   OK
          Ignore    No        Yes
rdpdr     Terminal Server Device Redirector Driver
          c:\windows\system32\drivers\rdpdr.sys Kernel Driver Yes
          Manual    Running   OK        Normal    No
          Yes
rdpwd     RDPWD     c:\windows\system32\drivers\rdpwd.sys Kernel
Driver    Yes       Manual    Running   OK        Ignore
          No        Yes
```

```
redbook     Digital CD Audio Playback Filter Driver
            c:\windows\system32\drivers\redbook.sys                 Kernel
Driver      Yes         System      Running     OK          Normal
            No          Yes
serial      Serial      c:\windows\system32\drivers\serial.sys      Kernel
Driver      No          Auto        Stopped     OK          Ignore
            No          No
sfloppy     Sfloppy     c:\windows\system32\drivers\sfloppy.sys
            Kernel Driver No        System      Stopped     OK
            Ignore      No          No
simbad      Simbad      Not Available Kernel Driver No      Disabled
            Stopped     OK          Normal      No          No

srv         Srv         c:\windows\system32\drivers\srv.sys         File
System Driver           Yes         Manual      Running     OK
            Normal      No          Yes
swenum      Software Bus Driver
            c:\windows\system32\drivers\swenum.sys                  Kernel
Driver      Yes         Manual      Running     OK          Normal
            No          Yes
symc8xx     symc8xx     Not Available Kernel Driver No      Disabled
            Stopped     OK          Normal      No          No

symmpi      symmpi      Not Available Kernel Driver No      Disabled
            Stopped     OK          Normal      No          No

sym_hi      sym_hi      c:\windows\system32\drivers\sym_hi.sys
            Kernel Driver Yes       Boot        Running     OK
            Normal      No          Yes
sym_u3      sym_u3      c:\windows\system32\drivers\sym_u3.sys
            Kernel Driver Yes       Boot        Running     OK
            Normal      No          Yes
tcpip       TCP/IP Protocol Driver
            c:\windows\system32\drivers\tcpip.sys  Kernel Driver Yes
            System      Running     OK          Normal      No
            Yes
tdpipe      TDPIPE      c:\windows\system32\drivers\tdpipe.sys Kernel
Driver      No          Manual      Stopped     OK          Ignore
            No          No
tdtcp       TDTCP       c:\windows\system32\drivers\tdtcp.sys       Kernel
Driver      Yes         Manual      Running     OK          Ignore
            No          Yes
termdd      Terminal Device Driver
            c:\windows\system32\drivers\termdd.sys                  Kernel
Driver      Yes         System      Running     OK          Normal
            No          Yes
toside      TosIde      Not Available Kernel Driver No      Disabled
            Stopped     OK          Normal      No          No

udfs        Udfs        c:\windows\system32\drivers\udfs.sys        File
System Driver           No          Disabled    Stopped     OK
            Normal      No          No

usbccgp     Microsoft USB Generic Parent Driver
            c:\windows\system32\drivers\usbccgp.sys                 Kernel
Driver      No          Manual      Stopped     OK          Normal
usbehci     Microsoft USB 2.0 Enhanced Host Controller Miniport Driver
            No          No
            c:\windows\system32\drivers\usbehci.sys                 Kernel
Driver      No          Manual      Stopped     OK          Normal
            No          No
usbhub      USB2 Enabled Hub
            c:\windows\system32\drivers\usbhub.sys                  Kernel
Driver      No          Manual      Stopped     OK          Normal
            No          No
usbohci     Microsoft USB Open Host Controller Miniport Driver
            c:\windows\system32\drivers\usbohci.sys                 Kernel
Driver      No          Manual      Stopped     OK          Normal
            No          No
vga         vga         c:\windows\system32\drivers\vgapnp.sys
            Kernel Driver No        Manual      Stopped     OK
            Ignore      No          No
vgasave     VGA Display Controller.
            c:\windows\system32\drivers\vga.sys  Kernel Driver No
            System      Stopped     OK          Ignore      No
            No
viaide      ViaIde      Not Available Kernel Driver No      Disabled
            Stopped     OK          Normal      No          No

volsnap     VolSnap     c:\windows\system32\drivers\volsnap.sys
            Kernel Driver Yes       Boot        Running     OK
            Normal      No          Yes
wanarp      Remote Access IP ARP Driver
            c:\windows\system32\drivers\wanarp.sys                  Kernel
Driver      Yes         Manual      Running     OK          Normal
            No          Yes
wdica       WDICA       Not Available Kernel Driver No      Manual
            Stopped     OK          Ignore      No          No

wlbs        Network Load Balancing
            c:\windows\system32\drivers\wlbs.sys  Kernel Driver No
            Manual      Stopped     OK          Normal      No
            No
ws2ifsl     Windows Socket 2.0 Non-IFS Service Provider Support
Environment c:\windows\system32\drivers\ws2ifsl.sysKernel Driver No
            Disabled    Stopped     OK          Normal      No
            No


[Signed Drivers]

Device Name Signed      Device Class Driver Version             Driver
Date        Manufacturer INF Name   Driver Name Device ID
Not Available Not Available Not Available Not Available Not Available Not
Available   Not Available Not Available HTREE\ROOT\0
ACPI IA64-based PC      No          COMPUTER    5.2.3663.0
            7/15/2002   (Standard computers)    hal.inf     Not
Available   ROOT\ACPI_HAL\0000

Microsoft ACPI-Compliant System     No          SYSTEM
            5.2.3663.0  7/15/2002   Microsoft   acpi.inf    Not
Available   ACPI_HAL\PNP0C08\0      No          SYSTEM      5.2.3663.0
            7/15/2002   (Standard system devices) machine.inf   Not
Available   ACPI\THERMALZONE\THM0
Processor   No          PROCESSOR   5.2.3663.0  7/15/2002   (Standard
processor types)        cpu.inf     Not Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_0\_0
Processor   No          PROCESSOR   5.2.3663.0  7/15/2002   (Standard
processor types)        cpu.inf     Not Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_0\_1
Processor   No          PROCESSOR   5.2.3663.0  7/15/2002   (Standard
processor types)        cpu.inf     Not Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_0\_2
Processor   No          PROCESSOR   5.2.3663.0  7/15/2002   (Standard
processor types)        cpu.inf     Not Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_0\_3
Generic Bus No          SYSTEM      5.2.3663.0  7/15/2002   (Standard
system devices)         machine.inf Not Available ACPI\HWP0001\0

Not Available Not Available UNKNOWN Not Available Not Available Not
Available   Not Available Not Available ACPI\IPI0001\0
PCI bus     No          SYSTEM      5.2.3663.0  7/15/2002   (Standard
system devices)         machine.inf Not Available ACPI\HWP0002\0

PCI Simple Communications Controller Not Available UNKNOWN  Not
Available   Not Available Not Available Not Available Not Available
            PCI\VEN_103C&DEV_1290&SUBSYS_1291103C&REV_01\4&4F5
EBC7&0&08
PCI Serial Port         Not Available UNKNOWN   Not Available Not
Available   Not Available Not Available Not Available
            PCI\VEN_103C&DEV_1048&SUBSYS_1282103C&REV_03\4&4F5
EBC7&0&09
LSI Logic 53C896 Device No          SCSIADAPTER
            5.2.3663.0  7/15/2002   LSI Logic Inc.
            pnpscsi.inf Not Available
            PCI\VEN_1000&DEV_000B&SUBSYS_00000000&REV_07\4&4F5
EBC7&0&10
LSI Logic 53C896 Device No          SCSIADAPTER
            5.2.3663.0  7/15/2002   LSI Logic Inc.
            pnpscsi.inf Not Available
            PCI\VEN_1000&DEV_000B&SUBSYS_00000000&REV_07\4&4F5
EBC7&0&11
CD-ROM Drive            No          CDROM       5.2.3663.0
            7/15/2002   (Standard CD-ROM drives) cdrom.inf      Not
Available   SCSI\CDROM&VEN_HP&PROD_DVD-
ROM_305&REV_1.01\5&968E4F&0&020
PCI bus     No          SYSTEM      5.2.3663.0  7/15/2002   (Standard
system devices)         machine.inf Not Available ACPI\HWP0002\100
```

PCI standard PCI-to-PCI bridge No SYSTEM
5.2.3663.0 7/15/2002 (Standard system devices)
machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\4&2C1
78B65&0&08
LSI Logic 53C1010-66 Device No SCSIADAPTER
5.2.3663.0 7/15/2002 LSI Logic Inc.
pnpscsi.inf Not Available
PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&2B7
E6A47&0&0808
LSI Logic 53C1010-66 Device No SCSIADAPTER
5.2.3663.0 7/15/2002 LSI Logic Inc.
pnpscsi.inf Not Available
PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&2B7
E6A47&0&0908
Disk drive No DISKDRIVE 5.2.3663.0 7/15/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_FUJITSU&PROD_MAJ3364MC&REV_HP12\6&3
840ADF6&0&000
Broadcom NetXtreme Gigabit Ethernet No NET 2.67.0.0
7/15/2002 Broadcom netb57xp.inf Not Available
PCI\VEN_14E4&DEV_1645&SUBSYS_1300103C&REV_15\5&2B7
E6A47&0&2008
PCI bus No SYSTEM 5.2.3663.0 7/15/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\200

PCI bus No SYSTEM 5.2.3663.0 7/15/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\300

QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 8/5/2002 QLogic
oem5.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&19E
BB955&0&08
Disk drive No DISKDRIVE 5.2.3663.0 7/15/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\5&26
6EAEBD&1&07F0
PCI bus No SYSTEM 5.2.3663.0 7/15/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\400

Smart Array 5300 Controller (Non-Miniport) No
SCSIADAPTER 5.5.58.64 9/17/2002 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\4&152
91AB&0&08
Smart Array Logical Volume No DISKDRIVE 5.5.54.64
8/15/2002 Hewlett-Packard oem4.inf Not
Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\5&17
F7BB1A&0&0000004000000000

Smart Array 5300 Controller (Non-Miniport) No
SCSIADAPTER 5.5.58.64 9/17/2002 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\4&152
91AB&0&10
Smart Array Logical Volume No DISKDRIVE 5.5.54.64
8/15/2002 Hewlett-Packard oem4.inf Not
Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\5&71
14797&0&0000004000000000
PCI bus No SYSTEM 5.2.3663.0 7/15/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\500

PCI bus No SYSTEM 5.2.3663.0 7/15/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\600

Smart Array 5300 Controller (Non-Miniport) No
SCSIADAPTER 5.5.58.64 9/17/2002 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\4&5D9
CB86&0&08
Smart Array Logical Volume No DISKDRIVE 5.5.54.64
8/15/2002 Hewlett-Packard oem4.inf Not
Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\5&1A
0C36B&0&0000004000000000
Smart Array 5300 Controller (Non-Miniport) No
SCSIADAPTER 5.5.58.64 9/17/2002 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\4&5D9
CB86&0&10
Smart Array Logical Volume No DISKDRIVE 5.5.55.64
9/17/2002 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\5&1E
328AC1&0&0000004000000000
PCI bus No SYSTEM 5.2.3663.0 7/15/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\700

Smart Array 5300 Controller (Non-Miniport) No
SCSIADAPTER 5.5.58.64 9/17/2002 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\4&1E7
2F330&0&08
Smart Array Logical Volume No DISKDRIVE 5.5.54.64
8/15/2002 Hewlett-Packard oem4.inf Not
Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\5&1
D6B2D68&0&0000004000000000
Smart Array 5300 Controller (Non-Miniport) No
SCSIADAPTER 5.5.58.64 9/17/2002 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\4&1E7
2F330&0&10
Smart Array Logical Volume No DISKDRIVE 5.5.55.64
9/17/2002 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\5&D
36612&0&0000004000000000
ACPI Fixed Feature Button SYSTEM 5.2.3663.0
7/15/2002 (Standard system devices) machine.inf Not
Available ACPI\FIXEDBUTTON\2&DABA3FF&0
Logical Disk Manager No SYSTEM 5.2.3663.0
7/15/2002 (Standard system devices) machine.inf Not
Available ROOT\DMIO\0000
Volume Manager No SYSTEM 5.2.3663.0
7/15/2002 (Standard system devices) machine.inf Not
Available ROOT\FTDISK\0000
Generic volume No VOLUME 5.2.3663.0
7/15/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{06699A9A
-0B2D-44DA-8D16-98AD22DAC1F2}
Generic volume No VOLUME 5.2.3663.0
7/15/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F9A189EF
-A36E-43C4-AB65-A268BC78F49E}
Generic volume No VOLUME 5.2.3663.0
7/15/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{54EA1F3F
-9B0D-4575-B2D9-ACBB8EC84CC7}
Generic volume No VOLUME 5.2.3663.0
7/15/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C89C0BCB
-F7CB-463E-8D1A-9A6AF10F4C4B}
Generic volume No VOLUME 5.2.3663.0
7/15/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{64D34861
-2E0C-4D13-9158-85864931F017}
Generic volume No VOLUME 5.2.3663.0
7/15/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{299C12D7
-E892-40D0-8DAE-D35C8A735C7E}
Generic volume No VOLUME 5.2.3663.0
7/15/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FC5C79E4
-14A2-4672-8CF6-62B7D334A48D}
Generic volume No VOLUME 5.2.3663.0
7/15/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{91B5792C
-35B1-48FD-9F5D-BA6383514389}
Generic volume No VOLUME 5.2.3663.0
7/15/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{67D4AC82
-104A-4997-94E9-CE3DA90E521B}
Generic volume No VOLUME 5.2.3663.0
7/15/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2AC0F4D
D-2476-4B9C-8583-2DF715D349DA}

Generic volume No VOLUME 5.2.3663.0 7/15/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{20B15B76-6AD2-43A9-8189-B6AFFE8CE7AF}
Generic volume No VOLUME 5.2.3663.0 7/15/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7A0BC1E8-7DBD-46C6-A57A-2BBE68B3EB29}
Generic volume No VOLUME 5.2.3663.0 7/15/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B1FB5D21-21F3-48DA-AFEF-BEF77CC23BF6}
Generic volume No VOLUME 5.2.3663.0 7/15/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{101D4F81-A9BD-4604-96BC-CA2B0E2CEFF6}
Generic volume No VOLUME 5.2.3663.0 7/15/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{37CC2663-7F6B-42C3-827E-6D46FB52093C}
Generic volume No VOLUME 5.2.3663.0 7/15/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{17D68EBA-5682-4BA5-94B9-96998FA81A8A}
Generic volume No VOLUME 5.2.3663.0 7/15/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7933BB6B-BBC7-4E9F-8456-08F2680BD054}
Generic volume No VOLUME 5.2.3663.0 7/15/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AC470066-8AD1-4BD3-B905-7953A5F1DB4A}
Generic volume No VOLUME 5.2.3663.0 7/15/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4B0A5C2A-0BA8-49A3-819F-9FC5F128D80C}
Generic volume No VOLUME 5.2.3663.0 7/15/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{88DCFBF4-7017-430B-8F0E-5A31F1AF663B}
Generic volume No VOLUME 5.2.3663.0 7/15/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9E2B4292-79B4-4B27-A551-534105FB4915}
Generic volume No VOLUME 5.2.3663.0 7/15/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C166540E-AB64-4FCE-9721-DC53BB7CE417}
Generic volume No VOLUME 5.2.3663.0 7/15/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FB95D260-479E-01C2-92E0-3C772E43AC40}

Generic volume No VOLUME 5.2.3663.0 7/15/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FBAB2F20-479E-01C2-967B-9E5F8078F531}
Generic volume No VOLUME 5.2.3663.0 7/15/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2B78BBA0-479F-01C2-F1B3-12714F758821}
AFD Networking Support Environment Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_AFD\0000
Beep Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_BEEP\0000
cpqcissm Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_CPQCISSM\0000
CRC Disk Filter Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_CRCDISK\0000
dac2w2k Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_DAC2W2K\0000
dmboot Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_DMBOOT\0000
dmload Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_DMLOAD\0000
em Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_EM\0000
Fips Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_FIPS\0000
Generic Packet Classifier Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_GPC\0000
hpn Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_HPN\0000
IPSEC driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_IPSEC\0000
ksecdd Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_KSECDD\0000
mountmgr Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_MOUNTMGR\0000
NDIS System Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NDIS\0000

Remote Access NDIS TAPI Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NDISTAPI\0000
NDIS Usermode I/O Protocol Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NDISUIO\0000
NDProxy Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_NDPROXY\0000
NetBios over Tcpip Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NETBT\0000
Null Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_NULL\0000
Partition Manager Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_PARTMGR\0000
qlvika Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_QLVIKA\0000
Remote Access Auto Connection Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_RASACD\0000
RDPCDD Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_RDPCDD\0000
RDPWD Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_RDPWD\0000
sacdrv Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_SACDRV\0000
TCP/IP Protocol Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_TCPIP\0000
TDTCP Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_TDTCP\0000
volsnap Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available ROOT\LEGACY_VOLSNAP\0000
Remote Access IP ARP Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_WANARP\0000
Windows Socket 2.0 Non-IFS Service Provider Support Environment Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_WS2IFSL\0000

Audio Codecs (Standard system devices) No MEDIA 5.2.3663.0 7/15/2002 wave.inf Not Available ROOT\MEDIA\MS_MMACM

```
Legacy Audio Drivers    No    MEDIA    5.2.3663.0
        7/15/2002    (Standard system devices)    wave.inf    Not
Available    ROOT\MEDIA\MS_MMDRV
Media Control Devices    No    MEDIA    5.2.3663.0
        7/15/2002    (Standard system devices)    wave.inf    Not
Available    ROOT\MEDIA\MS_MMMCI
Legacy Video Capture Devices    No    MEDIA
        5.2.3663.0    7/15/2002    (Standard system devices)    wave.inf
        Not Available    ROOT\MEDIA\MS_MMVCD
Video Codecs    No    MEDIA    5.2.3663.0    7/15/2002    (Standard
system devices)    wave.inf    Not Available
        ROOT\MEDIA\MS_MMVID
WAN Miniport (L2TP)    No    NET    5.2.3663.0
        7/15/2002    Microsoft    netrasa.inf    Not Available
        ROOT\MS_L2TPMINIPORT\0000
WAN Miniport (IP)    No    NET    5.2.3663.0
        7/15/2002    Microsoft    netrasa.inf    Not Available
        ROOT\MS_NDISWANIP\0000
WAN Miniport (PPPOE)    No    NET    5.2.3663.0
        7/15/2002    Microsoft    netrasa.inf    Not Available
        ROOT\MS_PPPOEMINIPORT\0000
WAN Miniport (PPTP)    No    NET    5.2.3663.0
        7/15/2002    Microsoft    netrasa.inf    Not Available
        ROOT\MS_PPTPMINIPORT\0000
Direct Parallel    No    NET    5.2.3663.0
        7/15/2002    Microsoft    netrasa.inf    Not Available
        ROOT\MS_PTIMINIPORT\0000
Terminal Server Device Redirector    No    SYSTEM
        5.2.3663.0    7/15/2002    (Standard system devices)
        machine.inf    Not Available    ROOT\RDPDR\0000
Terminal Server Keyboard Driver    No    SYSTEM
        5.2.3663.0    7/15/2002    (Standard system devices)
        machine.inf    Not Available    ROOT\RDP_KBD\0000
Terminal Server Mouse Driver    No    SYSTEM
        5.2.3663.0    7/15/2002    (Standard system devices)
        machine.inf    Not Available    ROOT\RDP_MOU\0000
Plug and Play Software Device Enumerator    No    SYSTEM
        5.2.3663.0    7/15/2002    (Standard system devices)
        machine.inf    Not Available    ROOT\SYSTEM\0000

[Environment Variables]

Variable    Value    User Name
_NT_SYMBOL_PATH    c:\Symbols    <SYSTEM>
ClusterLog    C:\WINDOWS\Cluster\cluster.log    <SYSTEM>
ComSpec    %SystemRoot%\system32\cmd.exe    <SYSTEM>
NUMBER_OF_PROCESSORS    4    <SYSTEM>
OS    Windows_NT    <SYSTEM>
Path
        %SystemRoot%\system32;%SystemRoot%;%SystemRoot%\Sy
stem32\Wbem;C:\Program Files\Microsoft SQL
Server\80\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL
Server\80\Tools\Binn\    <SYSTEM>
PATHEXT    .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
        <SYSTEM>
PROCESSOR_ARCHITECTURE    IA64    <SYSTEM>
PROCESSOR_IDENTIFIER    ia64 Family 31 Model 0 Stepping 6, GenuineIntel
        <SYSTEM>
PROCESSOR_LEVEL    31    <SYSTEM>
PROCESSOR_REVISION    0006    <SYSTEM>
TEMP    %SystemRoot%\TEMP    <SYSTEM>
TMP    %SystemRoot%\TEMP    <SYSTEM>
windir    %SystemRoot%    <SYSTEM>
TEMP    %USERPROFILE%\Local Settings\Temp    NT
AUTHORITY\SYSTEM
TMP    %USERPROFILE%\Local Settings\Temp    NT
AUTHORITY\SYSTEM
TEMP    %USERPROFILE%\Local Settings\Temp    NT AUTHORITY\LOCAL
SERVICE
TMP    %USERPROFILE%\Local Settings\Temp    NT AUTHORITY\LOCAL
SERVICE
TEMP    %USERPROFILE%\Local Settings\Temp    NT
AUTHORITY\NETWORK SERVICE
TMP    %USERPROFILE%\Local Settings\Temp    NT
AUTHORITY\NETWORK SERVICE
TEMP    %USERPROFILE%\Local Settings\Temp    TPC-
RX5670\Administrator
TMP    %USERPROFILE%\Local Settings\Temp    TPC-
RX5670\Administrator

[Print Jobs]

Document    Size    Owner    Notify    Status    Time
Submitted    Start Time    Until Time    Elapsed Time    Pages Printed
        Job ID    Priority    Parameters    Driver    Print
Processor    Host Print Queue    Data Type    Name

[Network Connections]

Local Name    Remote Name    Type    Status    User
Name
Y:    \\hptpc\c$    Disk    Persistent Connection    TPC-
RX5670\Administrator
Z:    \\ssdllab\d$    Disk    Persistent Connection    TPC-
RX5670\Administrator

[Running Tasks]

Name    Path    Process ID    Priority    Min Working Set
        Max Working Set    Start Time    Version    Size
        File Date
system idle process    Not Available    0    0    Not
Available    Not Available    Not Available    Not Available    Not
Available    Not Available    Not Available    Not Available
system    Not Available    4    8    0    2826240
Available    Not Available    Not Available    Not Available    Not
Available
smss.exe    c:\windows\system32\smss.exe    352    11
        409600    2826240    12/10/2002 9:57 AM
        5.2.3663.0 (main.020715-1506)    128.00 KB (131,072
bytes)    7/22/2002 5:00 AM
csrss.exe    Not Available    408    13    Not Available    Not
Available    12/10/2002 9:57 AM    Not Available    Not Available    Not
Available
winlogon.exe    c:\windows\system32\winlogon.exe    432    13
        409600    2826240    12/10/2002 9:57 AM
        5.2.3663.0 (main.020715-1506)    611.00 KB (625,664
bytes)    7/22/2002 5:00 AM
services.exe    c:\windows\system32\services.exe    476    9
        409600    2826240    12/10/2002 9:57 AM
        5.2.3663.0 (main.020715-1506)    292.50 KB (299,520
bytes)    7/22/2002 5:00 AM
lsass.exe    c:\windows\system32\lsass.exe    488    9
        409600    2826240    12/10/2002 9:57 AM
        5.2.3663.0 (main.020715-1506)    16.50 KB (16,896
bytes)    7/22/2002 5:00 AM
svchost.exe    c:\windows\system32\svchost.exe    640    8
        409600    2826240    12/10/2002 9:57 AM
        5.2.3663.0 (main.020715-1506)    32.50 KB (33,280
bytes)    7/22/2002 5:00 AM
svchost.exe    c:\windows\system32\svchost.exe    692    8
        409600    2826240    12/10/2002 9:57 AM
        5.2.3663.0 (main.020715-1506)    32.50 KB (33,280
bytes)    7/22/2002 5:00 AM
svchost.exe    Not Available    840    8    Not Available    Not
Available    12/10/2002 9:57 AM    Not Available    Not Available    Not
Available
svchost.exe    Not Available    860    8    Not Available    Not
Available    12/10/2002 9:57 AM    Not Available    Not Available    Not
Available
svchost.exe    c:\windows\system32\svchost.exe    940    8
        409600    2826240    12/10/2002 9:57 AM
        5.2.3663.0 (main.020715-1506)    32.50 KB (33,280
bytes)    7/22/2002 5:00 AM
msdtc.exe    Not Available    1020    8    Not Available    Not
Available    12/10/2002 9:57 AM    Not Available    Not Available    Not
Available
dllhost.exe    c:\windows\system32\dllhost.exe    1212    8
        409600    2826240    12/10/2002 9:57 AM
        5.2.3663.0 (main.020715-1506)    8.50 KB (8,704 bytes)
        7/22/2002 5:00 AM
dfssvc.exe    c:\windows\system32\dfssvc.exe    1356    8
        409600    2826240    12/10/2002 9:57 AM
        5.2.3663.0 (main.020715-1506)    413.00 KB (422,912
bytes)    7/22/2002 5:00 AM
```

wmiadap.exe \\?\c:\windows\system32\wbem\wmiadap.exe 1456 8 409600 2826240 12/10/2002 9:58 AM Not Available Not Available Not Available
csrss.exe Not Available 1708 13 Not Available Not Available 12/10/2002 9:58 AM Not Available Not Available Not Available
winlogon.exe c:\windows\system32\winlogon.exe 1736 13 409600 2826240 12/10/2002 9:58 AM 5.2.3663.0 (main.020715-1506) 611.00 KB (625,664 bytes) 7/22/2002 5:00 AM
rdpclip.exe c:\windows\system32\rdpclip.exe 1920 8 409600 2826240 12/10/2002 9:58 AM 5.2.3663.0 (main.020715-1506) 160.50 KB (164,352 bytes) 8/23/2002 1:13 PM
explorer.exe c:\windows\explorer.exe 184 8 409600 2826240 12/10/2002 9:58 AM 6.00.3663.0 (main.020715-1506) 1.64 MB (1,722,880 bytes) 7/22/2002 5:00 AM

helpctr.exe c:\windows\pchealth\helpctr\binaries\helpctr.exe 492 8 409600 2826240 12/10/2002 9:59 AM 5.2.3663.0 (main.020715-1506) 2.19 MB (2,298,368 bytes) 8/23/2002 1:16 PM
helpsvc.exe c:\windows\pchealth\helpctr\binaries\helpsvc.exe 904 8 409600 2826240 12/10/2002 9:59 AM 5.2.3663.0 (main.020715-1506) 2.43 MB (2,545,152 bytes) 8/23/2002 1:16 PM
wmiprvse.exe Not Available 1756 8 Not Available Not Available 12/10/2002 9:59 AM Not Available Not Available Not Available
wmiprvse.exe Not Available 1628 8 Not Available Not Available 12/10/2002 9:59 AM Not Available Not Available Not Available

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
smss	5.2.3663.0 (main.020715-1506)	128.00 KB (131,072 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\smss.exe
ntdll	5.2.3663.0 (main.020715-1506)	1.47 MB (1,539,072 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\ntdll.dll
winlogon	5.2.3663.0 (main.020715-1506)	611.00 KB (625,664 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\winlogon.exe
kernel32	5.2.3663.0 (main.020715-1506)	1.94 MB (2,039,296 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\kernel32.dll
msvcrt	7.0.3663.0 (main.020715-1506)	889.50 KB (910,848 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\msvcrt.dll
advapi32	5.2.3663.0 (main.020715-1506)	1.34 MB (1,403,904 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\advapi32.dll
rpcrt4	5.2.3663.0 (main.020715-1506)	2.09 MB (2,188,800 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\rpcrt4.dll
user32	5.2.3663.0 (main.020715-1506)	1.32 MB (1,387,008 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\user32.dll
gdi32	5.2.3663.0 (main.020715-1506)	802.50 KB (821,760 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\gdi32.dll
userenv	5.2.3663.0 (main.020715-1506)	1.48 MB (1,553,920 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\userenv.dll
nddeapi	5.2.3663.0 (main.020715-1506)	42.50 KB (43,520 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\nddeapi.dll
crypt32	5.131.3663.0 (main.020715-1506)	1.61 MB (1,683,968 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\crypt32.dll
msasn1	5.2.3663.0 (main.020715-1506)	160.00 KB (163,840 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\msasn1.dll
secur32	5.2.3663.0 (main.020715-1506)	170.50 KB (174,592 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\secur32.dll
winsta	5.2.3663.0 (main.020715-1506)	150.00 KB (153,600 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\winsta.dll
netapi32	5.2.3663.0 (main.020715-1506)	863.50 KB (884,224 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\netapi32.dll
profmap	5.2.3663.0 (main.020715-1506)	57.00 KB (58,368 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\profmap.dll
regapi	5.2.3663.0 (main.020715-1506)	129.50 KB (132,608 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\regapi.dll
ws2_32	5.2.3663.0 (main.020715-1506)	237.00 KB (242,688 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\ws2_32.dll
ws2help	5.2.3663.0 (main.020715-1506)	50.00 KB (51,200 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\ws2help.dll
authz	5.2.3663.0 (main.020715-1506)	191.00 KB (195,584 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\authz.dll
msgina	5.2.3663.0 (main.020715-1506)	2.01 MB (2,102,784 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\msgina.dll
shsvcs	6.00.3663.0 (main.020715-1506)	327.50 KB (335,360 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\shsvcs.dll
shlwapi	6.00.3663.0 (main.020715-1506)	749.50 KB (767,488 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\shlwapi.dll
sfc	5.2.3663.0 (main.020715-1506)	7.50 KB (7,680 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\sfc.dll
sfc_os	5.2.3663.0 (main.020715-1506)	251.00 KB (257,024 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\sfc_os.dll
wintrust	5.131.3663.0 (main.020715-1506)	467.00 KB (478,208 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\wintrust.dll
ole32	5.2.3663.0 (main.020715-1506)	3.71 MB (3,891,712 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\ole32.dll
imagehlp	5.2.3663.0 (main.020715-1506)	136.00 KB (139,264 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\imagehlp.dll
comctl32	6.0 (main.020715-1506)	2.26 MB (2,371,584 bytes)	8/23/2002 6:00 AM	Microsoft Corporation	c:\windows\winsxs\ia64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.100.0_x-ww_b3722bab\comctl32.dll
version	5.2.3663.0 (main.020715-1506)	44.00 KB (45,056 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\version.dll
winscard	5.2.3663.0 (main.020715-1506)	318.50 KB (326,144 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\winscard.dll
wtsapi32	5.2.3663.0 (main.020715-1506)	47.00 KB (48,128 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\wtsapi32.dll
winmm	5.2.3663.0 (main.020715-1506)	434.50 KB (444,928 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\winmm.dll
sxs	5.2.3663.0 (main.020715-1506)	1.95 MB (2,047,488 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\sxs.dll
shell32	6.00.3663.0 (main.020715-1506)	12.67 MB (13,283,328 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\shell32.dll
setupapi	5.2.3663.0 (main.020715-1506)	1.79 MB (1,880,576 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\setupapi.dll
wldap32	5.2.3663.0 (main.020715-1506)	390.50 KB (399,872 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\wldap32.dll
cscdll	5.2.3663.0 (main.020715-1506)	220.50 KB (225,792 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\cscdll.dll
wlnotify	5.2.3663.0 (main.020715-1506)	228.00 KB (233,472 bytes)	7/22/2002 5:00 AM	Microsoft Corporation	c:\windows\system32\wlnotify.dll

winspool 5.2.3663.0 (main.020715-1506) 424.50 KB (434,688 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\winspool.drv

mpr 5.2.3663.0 (main.020715-1506) 161.00 KB (164,864 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\mpr.dll

comctl32 5.82 (main.020715-1506) 1.55 MB (1,623,040 bytes) 8/23/2002 6:00 AM Microsoft Corporation c:\windows\winsxs\ia64_microsoft.windows.common-controls_6595b64144ccf1df_5.82.0.0_x-ww_b9c4a0a5\comctl32.dll

uxtheme 6.00.3663.0 (main.020715-1506) 549.50 KB (562,688 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\uxtheme.dll

scredir 5.2.3663.0 (main.020715-1506) 69.00 KB (70,656 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\scredir.dll

samlib 5.2.3663.0 (main.020715-1506) 111.00 KB (113,664 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\samlib.dll

clbcatq 2001.12.4593.0 (main.020715-1506) 1.44 MB (1,509,888 bytes) 8/23/2002 1:13 PM Microsoft Corporation c:\windows\system32\clbcatq.dll

oleaut32 5.2.3663.0 3.62 MB (3,792,896 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\oleaut32.dll

comres 2001.12.4593.0 (main.020715-1506) 779.50 KB (798,208 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\comres.dll

cscui 5.2.3663.0 (main.020715-1506) 628.50 KB (643,584 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\cscui.dll

drprov 5.2.3663.0 (main.020715-1506) 27.00 KB (27,648 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\drprov.dll

rdpsnd 5.2.3663.0 (main.020715-1506) 62.00 KB (63,488 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\rdpsnd.dll

ntlanman 5.2.3663.0 (main.020715-1506) 108.00 KB (110,592 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\ntlanman.dll

netui0 5.2.3663.0 (main.020715-1506) 187.00 KB (191,488 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\netui0.dll

netui1 5.2.3663.0 (main.020715-1506) 506.50 KB (518,656 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\netui1.dll

davclnt 5.2.3663.0 (main.020715-1506) 61.00 KB (62,464 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\davclnt.dll

mprui 5.2.3663.0 (main.020715-1506) 97.00 KB (99,328 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\mprui.dll

netui2 5.2.3663.0 (main.020715-1506) 796.00 KB (815,104 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\netui2.dll

comdlg32 6.00.3663.0 (main.020715-1506) 713.50 KB (730,624 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\comdlg32.dll

netmsg 5.2.3663.0 (main.020715-1506) 177.50 KB (181,760 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\netmsg.dll

ntmarta 5.2.3663.0 (main.020715-1506) 347.00 KB (355,328 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\ntmarta.dll

msacm32 5.2.3663.0 (main.020715-1506) 80.00 KB (81,920 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\msacm32.drv

msacm32 5.2.3663.0 (main.020715-1506) 252.50 KB (258,560 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\msacm32.dll

imaadp32 5.2.3663.0 (main.020715-1506) 55.00 KB (56,320 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\imaadp32.acm

msadp32 5.2.3663.0 (main.020715-1506) 48.50 KB (49,664 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\msadp32.acm

msg711 5.2.3663.0 (main.020715-1506) 33.00 KB (33,792 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\msg711.acm

msgsm32 5.2.3663.0 (main.020715-1506) 66.50 KB (68,096 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\msgsm32.acm

tssoft32 1.01 29.00 KB (29,696 bytes) 7/22/2002 5:00 AM DSP GROUP, INC. c:\windows\system32\tssoft32.acm

tsd32 Not Available 38.00 KB (38,912 bytes) 7/22/2002 5:00 AM Not Available c:\windows\system32\tsd32.dll

services 5.2.3663.0 (main.020715-1506) 292.50 KB (299,520 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\services.exe

scesrv 5.2.3663.0 (main.020715-1506) 840.00 KB (860,160 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\scesrv.dll

umpnpmgr 5.2.3663.0 (main.020715-1506) 319.50 KB (327,168 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\umpnpmgr.dll

ncobjapi 5.2.3663.0 (main.020715-1506) 119.50 KB (122,368 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\ncobjapi.dll

msvcp60 6.10.2154.8 941.50 KB (964,096 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\msvcp60.dll

eventlog 5.2.3663.0 (main.020715-1506) 158.00 KB (161,792 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\eventlog.dll

psapi 5.2.3663.0 (main.020715-1506) 48.00 KB (49,152 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\psapi.dll

lsass 5.2.3663.0 (main.020715-1506) 16.50 KB (16,896 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\lsass.exe

lsasrv 5.2.3663.0 (main.020715-1506) 1.95 MB (2,041,344 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\lsasrv.dll

samsrv 5.2.3663.0 (main.020715-1506) 1.04 MB (1,090,048 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\samsrv.dll

cryptdll 5.2.3663.0 (main.020715-1506) 62.50 KB (64,000 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\cryptdll.dll

dnsapi 5.2.3663.0 (main.020715-1506) 399.50 KB (409,088 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\dnsapi.dll

ntdsapi 5.2.3663.0 (main.020715-1506) 185.50 KB (189,952 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\ntdsapi.dll

msprivs 5.2.3663.0 (main.020715-1506) 43.50 KB (44,544 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\msprivs.dll

kerberos 5.2.3663.0 (main.020715-1506) 861.00 KB (881,664 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\kerberos.dll

msv1_0 5.2.3663.0 (main.020715-1506) 319.50 KB (327,168 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\msv1_0.dll

netlogon 5.2.3663.0 (main.020715-1506) 956.50 KB (979,456 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\netlogon.dll

w32time 5.2.3663.0 (main.020715-1506) 542.50 KB (555,520 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\w32time.dll

iphlpapi 5.2.3663.0 (main.020715-1506) 226.50 KB (231,936 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\iphlpapi.dll

schannel 5.2.3663.0 (main.020715-1506) 457.50 KB (468,480 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\schannel.dll

wdigest 5.2.3663.0 (main.020715-1506) 162.00 KB (165,888 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\wdigest.dll

rsaenh 5.2.3663.0 (main.020715-1506) 373.07 KB (382,024 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\rsaenh.dll

rassfm 5.2.3663.0 (main.020715-1506) 56.50 KB (57,856 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\rassfm.dll

kdcsvc 5.2.3663.0 (main.020715-1506) 568.50 KB (582,144 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\kdcsvc.dll

ntdsa	5.2.3663.0 (main.020715-1506)	3.64 MB (3,816,960 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\ntdsa.dll			
ntdsatq	5.2.3663.0 (main.020715-1506)	79.50 KB (81,408 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\ntdsatq.dll			
mswsock	5.2.3663.0 (main.020715-1506)	731.00 KB (748,544 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\mswsock.dll			
esent	5.2.3663.0 (main.020715-1506)	2.70 MB (2,826,752 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\esent.dll			
certcli	5.2.3663.0 (main.020715-1506)	585.50 KB (599,552 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\certcli.dll			
atl	3.00.2154	345.50 KB (353,792 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\atl.dll			
cryptui	5.131.3663.0 (main.020715-1506)	1.07 MB (1,117,696 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\cryptui.dll			
scecli	5.2.3663.0 (main.020715-1506)	497.50 KB (509,440 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\scecli.dll			
dssenh	5.2.3663.0 (main.020715-1506)	326.57 KB (334,408 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\dssenh.dll			
svchost	5.2.3663.0 (main.020715-1506)	32.50 KB (33,280 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\svchost.exe			
rpcss	5.2.3663.0 (main.020715-1506)	714.00 KB (731,136 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\rpcss.dll			
wshtcpip	5.2.3663.0 (main.020715-1506)	38.50 KB (39,424 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\wshtcpip.dll			
termsrv	5.2.3663.0 (main.020715-1506)	620.00 KB (634,880 bytes)	8/23/2002 1:13 PM	Microsoft Corporation
	c:\windows\system32\termsrv.dll			
icaapi	5.2.3663.0 (main.020715-1506)	26.50 KB (27,136 bytes)	8/23/2002 1:13 PM	Microsoft Corporation
	c:\windows\system32\icaapi.dll			
mstlsapi	5.2.3663.0 (main.020715-1506)	332.00 KB (339,968 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\mstlsapi.dll			
activeds	5.2.3663.0 (main.020715-1506)	574.00 KB (587,776 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\activeds.dll			
adsldpc	5.2.3663.0 (main.020715-1506)	314.00 KB (321,536 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\adsldpc.dll			
credui	5.2.3663.0 (main.020715-1506)	290.50 KB (297,472 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\credui.dll			
rdpwsx	5.2.3663.0 (main.020715-1506)	277.13 KB (283,784 bytes)	8/23/2002 1:13 PM	Microsoft Corporation
	c:\windows\system32\rdpwsx.dll			
schedsvc	5.2.3663.0 (main.020715-1506)	507.00 KB (519,168 bytes)	8/23/2002 1:15 PM	Microsoft Corporation
	c:\windows\system32\schedsvc.dll			
msidle	6.00.3663.0 (main.020715-1506)	9.00 KB (9,216 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\msidle.dll			
wkssvc	5.2.3663.0 (main.020715-1506)	305.50 KB (312,832 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\wkssvc.dll			
appmgmts	5.2.3663.0 (main.020715-1506)	397.50 KB (407,040 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\appmgmts.dll			
dmserver	5.2.3663.0 (main.020715-1506)	44.00 KB (45,056 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\dmserver.dll			
es	2001.12.4593.0 (main.020715-1506)	677.50 KB (693,760 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\es.dll			
pchsvc	5.2.3663.0 (main.020715-1506)	105.50 KB (108,032 bytes)	8/23/2002 1:16 PM	Microsoft Corporation
	c:\windows\pchealth\helpctr\binaries\pchsvc.dll			
srvsvc	5.2.3663.0 (main.020715-1506)	189.50 KB (194,048 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\srvsvc.dll			
sacsvr	5.2.3663.0 (main.020715-1506)	27.50 KB (28,160 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\sacsvr.dll			
wmisvc	5.2.3663.0 (main.020715-1506)	349.00 KB (357,376 bytes)	8/23/2002 1:13 PM	Microsoft Corporation
	c:\windows\system32\wbem\wmisvc.dll			
vssapi	5.2.3663.0 (main.020715-1506)	1.16 MB (1,220,096 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\vssapi.dll			
wbemcomn	5.2.3663.0 (main.020715-1506)	623.00 KB (637,952 bytes)	8/23/2002 1:13 PM	Microsoft Corporation
	c:\windows\system32\wbem\wbemcomn.dll			
sens	5.2.3663.0 (main.020715-1506)	91.50 KB (93,696 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\sens.dll			
browser	5.2.3663.0 (main.020715-1506)	156.50 KB (160,256 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\browser.dll			
netrap	5.2.3663.0 (main.020715-1506)	30.50 KB (31,232 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\netrap.dll			
wbemcore	5.2.3663.0 (main.020715-1506)	1.79 MB (1,882,112 bytes)	8/23/2002 1:13 PM	Microsoft Corporation
	c:\windows\system32\wbem\wbemcore.dll			
esscli	5.2.3663.0 (main.020715-1506)	957.00 KB (979,968 bytes)	8/23/2002 1:13 PM	Microsoft Corporation
	c:\windows\system32\wbem\esscli.dll			
fastprox	5.2.3663.0 (main.020715-1506)	1.58 MB (1,661,440 bytes)	8/23/2002 1:13 PM	Microsoft Corporation
	c:\windows\system32\wbem\fastprox.dll			
wbemsvc	5.2.3663.0 (main.020715-1506)	64.00 KB (65,536 bytes)	8/23/2002 1:13 PM	Microsoft Corporation
	c:\windows\system32\wbem\wbemsvc.dll			
wmiutils	5.2.3663.0 (main.020715-1506)	280.50 KB (287,232 bytes)	8/23/2002 1:13 PM	Microsoft Corporation
	c:\windows\system32\wbem\wmiutils.dll			
repdrvfs	5.2.3663.0 (main.020715-1506)	512.50 KB (524,800 bytes)	8/23/2002 1:13 PM	Microsoft Corporation
	c:\windows\system32\wbem\repdrvfs.dll			
netman	5.2.3663.0 (main.020715-1506)	501.50 KB (513,536 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\netman.dll			
mprapi	5.2.3663.0 (main.020715-1506)	262.00 KB (268,288 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\mprapi.dll			
rtutils	5.2.3663.0 (main.020715-1506)	86.50 KB (88,576 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\rtutils.dll			
rasapi32	5.2.3663.0 (main.020715-1506)	626.50 KB (641,536 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\rasapi32.dll			
rasman	5.2.3663.0 (main.020715-1506)	165.50 KB (169,472 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\rasman.dll			
tapi32	5.2.3663.0 (main.020715-1506)	517.50 KB (529,920 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\tapi32.dll			
wzcsvc	5.2.3663.0 (main.020715-1506)	609.00 KB (623,616 bytes)	7/16/2002 6:45 AM	Microsoft Corporation
	c:\windows\system32\wzcsvc.dll			
wmi	5.2.3663.0 (main.020715-1506)	5.00 KB (5,120 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\wmi.dll			
dhcpcsvc	5.2.3663.0 (main.020715-1506)	281.50 KB (288,256 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\dhcpcsvc.dll			
wzcsapi	5.2.3663.0 (main.020715-1506)	49.00 KB (50,176 bytes)	7/16/2002 6:45 AM	Microsoft Corporation
	c:\windows\system32\wzcsapi.dll			
netshell	5.2.3663.0 (main.020715-1506)	2.61 MB (2,738,176 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\netshell.dll			
clusapi	5.2.3663.0 (main.020715-1506)	170.00 KB (174,080 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\clusapi.dll			
hnetcfg	5.2.3663.0 (main.020715-1506)	795.50 KB (814,592 bytes)	7/22/2002 5:00 AM	Microsoft Corporation
	c:\windows\system32\hnetcfg.dll			

wininet 6.00.3663.0 (main.020715-1506) 1.56 MB (1,640,960 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\wininet.dll

rasdlg 5.2.3663.0 (main.020715-1506) 1.35 MB (1,420,288 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\rasdlg.dll

rasadhlp 5.2.3663.0 (main.020715-1506) 13.00 KB (13,312 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\rasadhlp.dll

wmiprvsd 5.2.3663.0 (main.020715-1506) 1.47 MB (1,542,656 bytes) 8/23/2002 1:13 PM Microsoft Corporation c:\windows\system32\wbem\wmiprvsd.dll

wbemess 5.2.3663.0 (main.020715-1506) 1,016.50 KB (1,040,896 bytes) 8/23/2002 1:13 PM Microsoft Corporation c:\windows\system32\wbem\wbemess.dll

ncprov 5.2.3663.0 (main.020715-1506) 137.50 KB (140,800 bytes) 8/23/2002 1:13 PM Microsoft Corporation c:\windows\system32\wbem\ncprov.dll

dllhost 5.2.3663.0 (main.020715-1506) 8.50 KB (8,704 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\dllhost.exe

comsvcs 2001.12.4593.0 (main.020715-1506) 3.31 MB (3,469,824 bytes) 8/23/2002 1:13 PM Microsoft Corporation c:\windows\system32\comsvcs.dll

mtxoci 2001.12.4593.0 (main.020715-1506) 320.50 KB (328,192 bytes) 8/23/2002 1:13 PM Microsoft Corporation c:\windows\system32\mtxoci.dll

txflog 2001.12.4593.0 (main.020715-1506) 311.00 KB (318,464 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\txflog.dll

dfssvc 5.2.3663.0 (main.020715-1506) 413.00 KB (422,912 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\dfssvc.exe

resutils 5.2.3663.0 (main.020715-1506) 152.50 KB (156,160 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\resutils.dll

wsock32 5.2.3663.0 (main.020715-1506) 23.00 KB (23,552 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\wsock32.dll

"\\?\c:\windows\system32\wbem\wmiadap.exe"

"\\?\c:\windows\system32\wbem\wmiadap.exe" "\\?\c:\windows\system32\wbem\wbemcomn.dll"

"\\?\c:\windows\system32\wbem\wbemcomn.dll"

loadperf 5.2.3663.0 (main.020715-1506) 230.50 KB (236,032 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\loadperf.dll

wbemprox 5.2.3663.0 (main.020715-1506) 46.00 KB (47,104 bytes) 8/23/2002 1:13 PM Microsoft Corporation c:\windows\system32\wbem\wbemprox.dll

rdpclip 5.2.3663.0 (main.020715-1506) 160.50 KB (164,352 bytes) 8/23/2002 1:13 PM Microsoft Corporation c:\windows\system32\rdpclip.exe

explorer 6.00.3663.0 (main.020715-1506) 1.64 MB (1,722,880 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\explorer.exe

browseui 6.00.3663.0 (main.020715-1506) 2.60 MB (2,728,960 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\browseui.dll

shdocvw 6.00.3663.0 (main.020715-1506) 3.33 MB (3,491,840 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\shdocvw.dll

apphelp 5.2.3663.0 (main.020715-1506) 265.50 KB (271,872 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\apphelp.dll

themeui 6.00.3663.0 (main.020715-1506) 835.50 KB (855,552 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\themeui.dll

msimg32 5.2.3663.0 (main.020715-1506) 7.00 KB (7,168 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\msimg32.dll

linkinfo 5.2.3663.0 (main.020715-1506) 44.00 KB (45,056 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\linkinfo.dll

ntshrui 6.00.3663.0 (main.020715-1506) 237.00 KB (242,688 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\ntshrui.dll

webcheck 6.00.3663.0 (main.020715-1506) 681.50 KB (697,856 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\webcheck.dll

stobject 5.2.3663.0 (main.020715-1506) 179.50 KB (183,808 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\stobject.dll

batmeter 6.00.3663.0 (main.020715-1506) 59.00 KB (60,416 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\batmeter.dll

powrprof 6.00.3663.0 (main.020715-1506) 38.50 KB (39,424 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\powrprof.dll

urlmon 6.00.3663.0 (main.020715-1506) 1.27 MB (1,326,592 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\urlmon.dll

printui 5.2.3663.0 (main.020715-1506) 1.13 MB (1,187,840 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\printui.dll

cfgmgr32 5.2.3663.0 (main.020715-1506) 15.50 KB (15,872 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\cfgmgr32.dll

helpctr 5.2.3663.0 (main.020715-1506) 2.19 MB (2,298,368 bytes) 8/23/2002 1:16 PM Microsoft Corporation c:\windows\pchealth\helpctr\binaries\helpctr.exe

hcappres 5.2.3663.0 (main.020715-1506) 6.00 KB (6,144 bytes) 8/23/2002 1:16 PM Microsoft Corporation c:\windows\pchealth\helpctr\binaries\hcappres.dll

itss 5.2.3663.0 (main.020715-1506) 352.00 KB (360,448 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\itss.dll

msxml3 8.40.8806.0 2.64 MB (3,814,912 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\msxml3.dll

pchshell 5.2.3663.0 (main.020715-1506) 306.50 KB (313,856 bytes) 8/23/2002 1:16 PM Microsoft Corporation c:\windows\pchealth\helpctr\binaries\pchshell.dll

mlang 6.00.3663.0 (main.020715-1506) 816.50 KB (836,096 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\mlang.dll

mshtml 6.00.3663.0 (main.020715-1506) 8.27 MB (8,676,352 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\mshtml.dll

msimtf 5.2.3663.0 (main.020715-1506) 562.50 KB (576,000 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\msimtf.dll

msctf 5.2.3663.0 (main.020715-1506) 985.00 KB (1,008,640 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\msctf.dll

shdoclc 6.00.3663.0 (main.020715-1506) 520.50 KB (532,992 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\shdoclc.dll

jscript 5.6.0.7727 1.24 MB (1,300,480 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\jscript.dll

msls31 3.10.349.0 444.50 KB (455,168 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\msls31.dll

imm32 5.2.3663.0 (main.020715-1506) 313.50 KB (321,024 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\imm32.dll

mshtmled 6.00.3663.0 (main.020715-1506) 1.42 MB (1,485,312 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\mshtmled.dll

vbscript 5.6.0.7727 1.09 MB (1,140,224 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\vbscript.dll

msinfo 5.2.3663.0 (main.020715-1506) 1.22 MB (1,282,048 bytes) 8/23/2002 1:16 PM Microsoft Corporation c:\windows\pchealth\helpctr\binaries\msinfo.dll

mfc42u 6.00.2178.0 3.34 MB (3,506,176 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\mfc42u.dll

riched32 5.2.3663.0 (main.020715-1506) 5.00 KB (5,120 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\riched32.dll

riched20 5.31.23.1217 1.27 MB (1,327,616 bytes) 7/22/2002 5:00 AM Microsoft Corporation c:\windows\system32\riched20.dll

helpsvc 5.2.3663.0 (main.020715-1506) 2.43 MB (2,545,152 bytes) 8/23/2002 1:16 PM Microsoft Corporation c:\windows\pchealth\helpctr\binaries\helpsvc.exe

[Services]

Display Name	Name	State	Start Mode	Service Type	Path	Error Control	Start Name	Tag ID
Alerter	Alerter	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k localservice	Normal	NT AUTHORITY\LocalService	0
Application Layer Gateway Service	ALG	Stopped	Manual	Own Process	c:\windows\system32\alg.exe	Normal	NT AUTHORITY\LocalService	0
Application Management	AppMgmt	Running	Auto	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
Windows Audio	AudioSrv	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
Background Intelligent Transfer Service	BITS	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
Computer Browser	Browser	Running	Auto	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
Indexing Service	CiSvc	Stopped	Manual	Share Process	c:\windows\system32\cisvc.exe	Normal	LocalSystem	0
ClipBook	ClipSrv	Stopped	Disabled	Own Process	c:\windows\system32\clipsrv.exe	Normal	LocalSystem	0
COM+ System Application	COMSysApp	Running	Auto	Own Process	c:\windows\system32\dllhost.exe /processid:{02d4b3f1-fd88-11d1-960d-00805fc79235}	Normal	LocalSystem	0
Cryptographic Services	CryptSvc	Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
Distributed File System	Dfs	Running	Auto	Own Process	c:\windows\system32\dfssvc.exe	Normal	LocalSystem	0
DHCP Client	Dhcp	Running	Auto	Share Process	c:\windows\system32\svchost.exe -k networkservice	Normal	NT AUTHORITY\NetworkService	0
Logical Disk Manager Administrative Service	dmadmin	Stopped	Manual	Share Process	c:\windows\system32\dmadmin.exe /com	Normal	LocalSystem	0
Logical Disk Manager	dmserver	Running	Auto	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
DNS Client	Dnscache	Running	Auto	Share Process	c:\windows\system32\svchost.exe -k networkservice	Normal	NT AUTHORITY\NetworkService	0
Error Reporting Service	ERSvc	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Ignore	LocalSystem	0
Event Log	Eventlog	Running	Auto	Share Process	c:\windows\system32\services.exe	Normal	LocalSystem	0
COM+ Event System	EventSystem	Running	Auto	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
Help and Support	helpsvc	Running	Auto	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
Human Interface Device Access	HidServ	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
HTTP SSL	HTTPFilter	Stopped	Manual	Share Process	c:\windows\system32\lsass.exe	Normal	LocalSystem	0
IAS Jet Database Access	IASJet	Stopped	Manual	Share Process	c:\windows\syswow64\svchost.exe -k iasjet	Normal	LocalSystem	0
IMAPI CD-Burning COM Service	ImapiService	Stopped	Disabled	Own Process	"c:\windows\system32\imapi.exe"	Normal	LocalSystem	0
Intersite Messaging	IsmServ	Stopped	Disabled	Own Process	c:\windows\system32\ismserv.exe	Normal	LocalSystem	0
Kerberos Key Distribution Center	kdc	Stopped	Disabled	Share Process	c:\windows\system32\lsass.exe	Normal	LocalSystem	0
Server	lanmanserver	Running	Auto	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
Workstation	lanmanworkstation	Running	Auto	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
License Logging	LicenseService	Stopped	Disabled	Own Process	c:\windows\system32\llssrv.exe	Normal	NT AUTHORITY\NetworkService	0
TCP/IP NetBIOS Helper	LmHosts	Running	Auto	Share Process	c:\windows\system32\svchost.exe -k localservice	Normal	NT AUTHORITY\LocalService	0
Machine Debug Manager	MDM	Stopped	Disabled	Own Process	"c:\w64\bin\win64\debugger\mdm.exe"	Normal	LocalSystem	0
Messenger	Messenger	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
Distributed Transaction Coordinator	MSDTC	Running	Auto	Own Process	c:\windows\system32\msdtc.exe	Normal	NT AUTHORITY\NetworkService	0
Windows Installer	MSIServer	Stopped	Manual	Share Process	c:\windows\system32\msiexec.exe /v	Normal	LocalSystem	0
MSSQLSERVER	MSSQLSERVER	Stopped	Manual	Own Process	c:\program files\microsoft sql server\mssql\binn\sqlservr.exe -smssqlserver	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	Disabled	Share Process	c:\windows\system32\netdde.exe	Normal	LocalSystem	0
Network DDE DSDM	NetDDEdsdm	Stopped	Disabled	Share Process	c:\windows\system32\netdde.exe	Normal	LocalSystem	0
Net Logon	Netlogon	Stopped	Manual	Share Process	c:\windows\system32\lsass.exe	Normal	LocalSystem	0
Network Connections	Netman	Running	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
Network Location Awareness (NLA)	Nla	Running	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
File Replication	NtFrs	Stopped	Manual	Own Process	c:\windows\system32\ntfrs.exe	Ignore	LocalSystem	0
NT LM Security Support Provider	NtLmSsp	Stopped	Manual	Share Process	c:\windows\system32\lsass.exe	Normal	LocalSystem	0
Removable Storage	NtmsSvc	Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
Plug and Play	PlugPlay	Running	Auto	Share Process	c:\windows\system32\services.exe	Normal	LocalSystem	0
IPSEC Services	PolicyAgent	Stopped	Disabled	Share Process	c:\windows\system32\lsass.exe	Normal	LocalSystem	0
Protected Storage	ProtectedStorage	Stopped	Manual	Share Process	c:\windows\system32\lsass.exe	Normal	LocalSystem	0
Remote Access Auto Connection Manager	RasAuto	Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
Remote Access Connection Manager	RasMan	Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
Remote Desktop Help Session Manager	RDSessMgr	Stopped	Manual	Own Process	c:\windows\system32\sessmgr.exe	Normal	LocalSystem	0

```
Routing and Remote Access RemoteAccess Stopped
        Disabled Share Process
        c:\windows\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
Remote Registry RemoteRegistry Stopped Manual
        Share Process c:\windows\system32\svchost.exe -k
regsvc Normal NT AUTHORITY\LocalService 0

Remote Procedure Call (RPC) Locator RpcLocator Stopped Manual
        Own Process c:\windows\system32\locator.exe Normal
        NT AUTHORITY\NetworkService 0
Remote Procedure Call (RPC) RpcSs Running Auto
        Share Process c:\windows\system32\svchost -k
rpcss Normal LocalSystem 0
Resultant Set of Policy Provider RSoPProv Stopped Manual
        Share Process c:\windows\system32\rsopprov.exe
        Normal LocalSystem 0
Special Administration Console Helper sacsvr Running Manual
        Share Process c:\windows\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Security Accounts Manager SamSs Running Auto Share
Process c:\windows\system32\lsass.exe Normal
        LocalSystem 0
Smart Card SCardSvr Stopped Manual Share Process
        c:\windows\system32\scardsvr.exe Ignore NT
AUTHORITY\LocalService 0
Task Scheduler Schedule Running Auto Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
Secondary Logon seclogon Stopped Disabled Share
Process c:\windows\system32\svchost.exe -k netsvcs Ignore
        LocalSystem 0
System Event Notification SENS Running Auto Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
Shell Hardware Detection ShellHWDetection Stopped Disabled
        Share Process c:\windows\system32\svchost.exe -k
netsvcs Ignore LocalSystem 0
Print Spooler Spooler Stopped Disabled Own Process
        c:\windows\system32\spoolsv.exe Normal
        LocalSystem 0
SQLSERVERAGENT SQLSERVERAGENT Stopped Manual
        Own Process c:\program files\microsoft sql
server\mssql\binn\sqlagent.exe -i mssqlserver Normal
        LocalSystem 0
Windows Image Acquisition (WIA) stisvc Stopped Disabled
        Share Process c:\windows\system32\svchost.exe -k
imgsvc Normal NT AUTHORITY\LocalService 0

Microsoft Software Shadow Copy Provider swprv Stopped
        Manual Own Process c:\windows\system32\svchost.exe -k
swprv Normal LocalSystem 0
```

```
Performance Logs and Alerts SysmonLog Stopped Manual
        Own Process c:\windows\system32\smlogsvc.exe Normal
Telephony TapiSrv Stopped Manual Share Process
        NT Authority\NetworkService
        c:\windows\system32\svchost.exe -k tapisrv Normal
        LocalSystem 0
Terminal Services TermService Running Manual Share
Process c:\windows\system32\svchost.exe -k termsvcs Normal
        LocalSystem 0
Telnet TlntSvr Stopped Disabled Own Process
        c:\windows\system32\tlntsvr.exe Normal NT
AUTHORITY\LOCAL SERVICE 0
Distributed Link Tracking Server TrkSvr Stopped Disabled
        Share Process c:\windows\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Distributed Link Tracking Client TrkWks Stopped Disabled
        Share Process c:\windows\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Terminal Services Session Directory Tssdis Stopped Disabled
        Own Process c:\windows\system32\tssdis.exe Normal
        LocalSystem 0
Upload Manager uploadmgr Stopped Disabled Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
Uninterruptible Power Supply UPS Stopped Manual
        Own Process c:\windows\system32\ups.exe Normal
        NT AUTHORITY\LocalService 0
Virtual Disk Service vds Stopped Manual Own
Process c:\windows\system32\vds.exe Normal
        LocalSystem 0
Volume Shadow Copy VSS Stopped Manual Own
Process c:\windows\system32\vssvc.exe Normal
        LocalSystem 0
Windows Time W32Time Stopped Disabled Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
WebClient WebClient Stopped Disabled Share Process
        c:\windows\system32\svchost.exe -k localservice Normal
        NT AUTHORITY\LocalService 0
WinHTTP Web Proxy Auto-Discovery Service WinHttpAutoProxySvc
        Stopped Manual Share Process
        c:\windows\system32\svchost.exe -k localservice Normal
        NT AUTHORITY\LocalService 0
Windows Management Instrumentation winmgmt Running Auto
        Share Process c:\windows\system32\svchost.exe -k
netsvcs Ignore LocalSystem 0
Windows Management Instrumentation Driver Extensions Wmi
        Stopped Manual Share Process
        c:\windows\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
WMI Performance Adapter WmiApSrv Stopped Manual Own
Process c:\windows\system32\wbem\wmiapsrv.exe Normal
        LocalSystem 0
```

```
Automatic Updates wuauserv Stopped Disabled Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
Wireless Configuration WZCSVC Stopped Disabled Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0

[Program Groups]

Group Name Name User Name
Accessories Default User:Accessories Default User
Accessories\Accessibility Default User:Accessories\Accessibility Default
User
Accessories\Entertainment Default User:Accessories\Entertainment Default
User
Startup Default User:Startup Default User
Accessories All Users:Accessories All Users
Accessories\Accessibility All Users:Accessories\Accessibility All Users

Accessories\Communications All
Users:Accessories\Communications All Users
Accessories\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 NT AUTHORITY\SYSTEM:Accessories NT
AUTHORITY\SYSTEM
Accessories\Accessibility NT AUTHORITY\SYSTEM:Accessories\Accessibility
        NT AUTHORITY\SYSTEM
Accessories\Entertainment NT
AUTHORITY\SYSTEM:Accessories\Entertainment NT
AUTHORITY\SYSTEM
Startup NT AUTHORITY\SYSTEM:Startup NT
AUTHORITY\SYSTEM
Accessories TPC-RX5670\Administrator:Accessories TPC-
RX5670\Administrator
Accessories\Accessibility TPC-
RX5670\Administrator:Accessories\Accessibility TPC-
RX5670\Administrator
Accessories\Entertainment TPC-
RX5670\Administrator:Accessories\Entertainment TPC-
RX5670\Administrator
Administrative Tools TPC-RX5670\Administrator:Administrative Tools
        TPC-RX5670\Administrator
SANblade Control VIX TPC-RX5670\Administrator:SANblade Control VIX
        TPC-RX5670\Administrator
Startup TPC-RX5670\Administrator:Startup TPC-
RX5670\Administrator

[Startup Programs]
```

Program	Command	User Name	Location
desktop	desktop.ini	NT AUTHORITY\SYSTEM	Startup
desktop	desktop.ini	TPC-RX5670\Administrator	Startup
desktop	desktop.ini	.DEFAULT	Startup
desktop	desktop.ini	All Users	Common Startup

[OLE Registration]

Object	Local Server
Sound (OLE2)	sndrec32.exe
Media Clip	mplay32.exe
Video Clip	mplay32.exe /avi
MIDI Sequence	mplay32.exe /mid
Sound	Not Available
Media Clip	Not Available
WordPad Document	"%programfiles%\windows nt\accessories\wordpad.exe"
Bitmap Image	mspaint.exe

[Windows Error Reporting]

Time	Type	Details

Microsoft SQL Server 8.0 Startup Parameters

sqlservr -x -c -T3502

where

-x	Disable the keeping of CPU time and cache-hit ratios.
-c	Start SQLServer independently of the Microsoft Windows 2000 Service Control Manager.
-T3502	Prints a message to the log at the beginning and end of each checkpoint.

BOOT.INI

One switch was added to the boot.ini file
/configflag=8224: interrupt logging disable.

User Rights Assignment

The Group Policy Editor of Windows.net was used to modify an entry under User Rights Assignment. Specifically, the right to "Lock pages in memory" was given to the Administrators group so that SQL Server 2000 could use large amounts of physical memory.

C.1 Microsoft SQL Server 8.0 Configuration Parameters

name	minimum	maximum	config_value	run_value
affinity mask	-2147483648	2147483647	0	0
affinity64 mask	-2147483648	2147483647	0	0
allow updates	0	1	0	0
awe enabled	0	1	0	0
c2 audit mode	0	1	0	0
cost threshold for parallelism	0	32767	5	5
Cross DB Ownership Chaining	0	1	0	0
cursor threshold	-1	2147483647	-1	-1
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	47000	47000
max text repl size (B)	0	2147483647	65536	65536
max worker threads	32	32767	260	260
media retention	0	365	0	0
min memory per query (KB)	512	2147483647	1024	1024
min server memory (MB)	0	2147483647	0	0
nested triggers	0	1	1	1
network packet size (B)	512	65536	4096	4096
open objects	0	2147483647	0	0
priority boost	0	1	1	1
query governor cost limit	0	2147483647	0	0
query wait (s)	-1	2147483647	-1	-1
recovery interval (min)	0	32767	56	56
remote access	0	1	1	1
remote login timeout (s)	0	2147483647	20	20
remote proc trans	0	1	0	0
remote query timeout (s)	0	2147483647	600	600
set working set size	0	1	0	0
scan for startup procs	0	1	0	0
show advanced options	0	1	1	1
two digit year cutoff	1753	9999	2049	2049
user connections	0	32767	0	0
user options	0	32767	0	0

C.2 Client System Configuration Parameters

COM+ Settings

TPCC.AllTxns:

Activation:

Enable Object Pooling selected
Minimum Pool Size: 45
Maximum Pool Size: 45
Creation Timeout: 60000
Enable Object Construction
Enable Just In Time Activation

Concurrency:

Concurrency Required

Microsoft IIS Registry Parameters

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000100
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,00,00
"PoolThreadLimit"=dword:000007d0
"ThreadTimeout"=dword:00015180
"MaxConnections"=dword:00004e20

World Wide Web Service Registry Parameters

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters
Class Name: <NO CLASS>
Last Write Time: 5/10/2002 - 2:37 PM
Value 0
Name: AcceptExOutstanding
Type: REG_DWORD
Data: 0x28

Value 1
Name: AccessDeniedMessage
Type: REG_SZ
Data: Error: Access is Denied.

Value 2
Name: CertMapList
Type: REG_SZ
Data: C:\WINNT\System32\inetsrv\iiscrmap.dll

Value 3
Name: Filter DLLs
Type: REG_SZ
Data:

Value 4
Name: InstallPath
Type: REG_SZ
Data: C:\WINNT\System32\inetsrv

Value 5
Name: LogFileDirectory
Type: REG_SZ
Data: C:\WINNT\System32\LogFiles

Value 6
Name: MajorVersion
Type: REG_DWORD
Data: 0x5

Value 7
Name: MinorVersion
Type: REG_DWORD
Data: 0

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch
Class Name: <NO CLASS>
Last Write Time: 5/6/2002 - 9:05 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory
Class Name: <NO CLASS>
Last Write Time: 5/6/2002 - 9:05 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory
Class Name: <NO CLASS>
Last Write Time: 5/6/2002 - 9:05 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map
Class Name: <NO CLASS>
Last Write Time: 5/7/2002 - 9:42 AM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots
Class Name: <NO CLASS>
Last Write Time: 6/28/2002 - 5:33 PM
Value 0
Name: /
Type: REG_SZ
Data: c:\inetpub\wwwroot,,205

Value 1
Name: /_vti_bin
Type: REG_SZ
Data: C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\40\isapi,,1

Value 2
Name: /IISAdmin
Type: REG_SZ
Data: C:\WINNT\System32\inetsrv\iisadmin,,1

Value 3
Name: /IISHelp
Type: REG_SZ
Data: c:\winnt\help\iishelp,,1

Value 4
Name: /IISSamples
Type: REG_SZ
Data: c:\inetpub\iissamples,,1

Value 5
Name: /MSADC
Type: REG_SZ
Data: c:\program files\common files\system\msadc,,1

Value 6
Name: /Printers
Type: REG_SZ
Data: C:\WINNT\web\printers,,201

Value 7
Name: /Scripts
Type: REG_SZ
Data: c:\inetpub\scripts,,1

TPCC Application Registry Parameters

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPCC]
"Path"="C:\\Inetpub\\wwwroot\\"
"NumberOfDeliveryThreads"=dword:00000008
"MaxConnections"=dword:000055f0
"MaxPendingDeliveries"=dword:000007d0
"DB_Protocol"="ODBC"
"TxnMonitor"="COM"
"DbServer"="tpc-rx5670"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"="****"
"COM_SinglePool"="YES"

TCPIP Registry Parameters

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters]
"NV Hostname"="tpcweb1"
"DataBasePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,\

00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,\
64,00,72,00,69,00,76,00,65,00,72,00,73,00,5c,00,65,00,74,00,63,00,00,00
"NameServer"=""
"ForwardBroadcasts"=dword:00000000
"IPEnableRouter"=dword:00000000
"Hostname"="tpcweb1"
"Domain"=""
"SearchList"=""
"UseDomainNameDevolution"=dword:00000001
"EnableICMPRedirect"=dword:00000001
"DeadGWDetectDefault"=dword:00000001
"DontAddDefaultGatewayDefault"=dword:00000000
"EnableSecurityFilters"=dword:00000000
"AllowUnqualifiedQuery"=dword:00000000
"PrioritizeRecordData"=dword:00000001
"MaxUserPort"=dword:0000fffe
"DhcpNameServer"="15.1.100.1"
"DhcpDomain"="hp-perf.net"

Microsoft Windows 2000 Server Configuration Parameters

System Information report written at: 12/09/2002 04:01:09 PM
[System Information]

[Following are sub-categories of this main category]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Server
Version	5.0.2195 Service Pack 3 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	TPCWEB1
System Manufacturer	Compaq
System Model	ProLiant DL360 G2
System Type	X86-based PC
Processor	x86 Family 6 Model 11 Stepping 4 GenuineIntel ~1396 Mhz
Processor	x86 Family 6 Model 11 Stepping 4 GenuineIntel ~1396 Mhz
BIOS Version	06/25/02
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition2
Locale	United States
User Name	TPCWEB1\Administrator
Time Zone	Pacific Standard Time
Total Physical Memory	1,048,088 KB
Available Physical Memory	861,616 KB
Total Virtual Memory	3,569,336 KB
Available Virtual Memory	3,298,588 KB
Page File Space	2,521,248 KB
Page File	C:\pagefile.sys

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

Resource	Device
IRQ 7	Standard OpenHCD USB Host Controller
IRQ 7	PCI standard host CPU bridge

[DMA]

Channel	Device	Status
7	Direct memory access controller	OK
2	Standard floppy disk controller	OK

[Forced Hardware]

Device	PNP Device ID
No Forced Hardware	

[I/O]

Address Range	Device	Status
0x0000-0x0CFF	PCI bus	OK
0x0000-0x0CFF	PCI bus	OK
0x0000-0x0CFF	Direct memory access controller	OK
0x03B0-0x03DF	PCI bus	OK
0x03B0-0x03DF	ATI Technologies Inc. RAGE XL PCI	OK
0x2400-0x24FF	ATI Technologies Inc. RAGE XL PCI	OK
0x03C0-0x03DF	ATI Technologies Inc. RAGE XL PCI	OK
0x1800-0x18FF	Compaq iLO Advanced System Management Controller	OK
0x2800-0x28FF	Compaq Integrated Lights-Out	OK
0x0A79-0x0A79	ISAPNP Read Data Port	OK
0x0279-0x0279	ISAPNP Read Data Port	OK
0x02F4-0x02F7	ISAPNP Read Data Port	OK
0x0F50-0x0F58	Motherboard resources	OK
0x0020-0x0021	Programmable interrupt controller	OK
0x00A0-0x00A1	Programmable interrupt controller	OK
0x0C00-0x0C01	Programmable interrupt controller	OK
0x0040-0x0043	System timer	OK
0x0080-0x008F	Direct memory access controller	OK
0x00C0-0x00DF	Direct memory access controller	OK
0x040B-0x040B	Direct memory access controller	OK
0x04D6-0x04D6	Direct memory access controller	OK
0x0061-0x0061	System speaker	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
0x002E-0x002F	Extended IO Bus	OK
0x0220-0x0223	Extended IO Bus	OK

```
0x0230-0x0231   Extended IO Bus   OK
0x0240-0x025F   Extended IO Bus   OK
0x03F8-0x03FF   Communications Port (COM1)   OK
0x03F2-0x03F5   Standard floppy disk controller   OK
0x03F7-0x03F7   Standard floppy disk controller   OK
0x2000-0x200F   Standard Dual Channel PCI IDE Controller
        OK
0x27FC-0x27FF   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
0x3000-0x30FF   PCI bus   OK
0x3000-0x30FF   Compaq Smart Array 5i   OK
0x4000-0x40FF   PCI bus   OK
0x4000-0x40FF   QLogic QLA23xx PCI Fibre Channel Adapter
        OK

[IRQs]

IRQ Number  Device
9           Microsoft ACPI-Compliant System
24          ATI Technologies Inc. RAGE XL PCI
23          Compaq iLO Advanced System Management Controller
22          Compaq Integrated Lights-Out
1           Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
12          PS/2 Compatible Mouse
4           Communications Port (COM1)
6           Standard floppy disk controller
14          Primary IDE Channel
7           Standard OpenHCD USB Host Controller
7           PCI standard host CPU bridge
31          Compaq Smart Array 5i
30          Compaq NC7780 Gigabit Server Adapter
29          Compaq NC7780 Gigabit Server Adapter #2
26          QLogic QLA23xx PCI Fibre Channel Adapter

[Memory]

Range   Device   Status
0xA0000-0xBFFFF   PCI bus   OK
0xA0000-0xBFFFF   ATI Technologies Inc. RAGE XL PCI   OK
0xF5E00000-0xF6FFFFFF   PCI bus   OK
0xF6000000-0xF6FFFFFF   ATI Technologies Inc. RAGE XL PCI   OK
0xF5FF0000-0xF5FF0FFF   ATI Technologies Inc. RAGE XL PCI   OK
0xF5FE0000-0xF5FE01FF   Compaq iLO Advanced System Management
Controller   OK
0xF5FD0000-0xF5FD07FF   Compaq Integrated Lights-Out   OK
0xF5FC0000-0xF5FC1FFF   Compaq Integrated Lights-Out   OK
0xF5F00000-0xF5F7FFFF   Compaq Integrated Lights-Out   OK
0xF5EF0000-0xF5EF0FFF   Standard OpenHCD USB Host Controller   OK
0xF7E00000-0xF7EFFFFF   PCI bus   OK
0xF7DF0000-0xF7DF3FFF   Compaq Smart Array 5i   OK
0xF7EB0000-0xF7EBFFFF   Compaq NC7780 Gigabit Server Adapter   OK
0xF7EA0000-0xF7EAFFFF   Compaq NC7780 Gigabit Server Adapter #2
        OK
0xF7F00000-0xF7FFFFFF   PCI bus   OK
0xF7FF0000-0xF7FF0FFF   QLogic QLA23xx PCI Fibre Channel Adapter
        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)   10/31/2002 2:28:37
AM
c:\winnt\system32\lhacm.acm   Microsoft Corporation
        OK   C:\WINNT\System32\LHACM.ACM   4.4.3385
        33.27 KB (34,064 bytes)   10/31/2002 10:46:40 AM
c:\winnt\system32\msg711.acm   Microsoft Corporation
        OK   C:\WINNT\System32\MSG711.ACM
        5.00.2134.1   10.27 KB (10,512 bytes)   10/31/2002 2:23:03
AM
c:\winnt\system32\msg723.acm   Microsoft Corporation
        OK   C:\WINNT\System32\MSG723.ACM   4.4.3385
        106.77 KB (109,328 bytes)   10/31/2002 10:46:40 AM
c:\winnt\system32\tssoft32.acm   DSP GROUP, INC.
        OK   C:\WINNT\System32\TSSOFT32.ACM   1.01
        9.27 KB (9,488 bytes)   10/31/2002 2:26:50 AM
c:\winnt\system32\msadp32.acm   Microsoft Corporation
        OK   C:\WINNT\System32\MSADP32.ACM
        5.00.2134.1   14.77 KB (15,120 bytes)   10/31/2002 2:22:56
AM
c:\winnt\system32\imaadp32.acm   Microsoft Corporation
        OK   C:\WINNT\System32\IMAADP32.ACM
        5.00.2134.1   16.27 KB (16,656 bytes)   10/31/2002 2:21:23
AM
c:\winnt\system32\msgsm32.acm   Microsoft Corporation
        OK   C:\WINNT\System32\MSGSM32.ACM
        5.00.2134.1   22.27 KB (22,800 bytes)   10/31/2002 2:23:03
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)
        10/31/2002 2:28:39 AM
c:\winnt\system32\msh261.drv   Microsoft Corporation
        OK   C:\WINNT\System32\MSH261.DRV   4.4.3385
        163.77 KB (167,696 bytes)   10/31/2002 10:46:40 AM
c:\winnt\system32\msvidc32.dll   Microsoft Corporation
        OK   C:\WINNT\System32\MSVIDC32.DLL
        5.00.2134.1   27.27 KB (27,920 bytes)   10/31/2002 2:23:27
AM
c:\winnt\system32\msrle32.dll   Microsoft Corporation
        OK   C:\WINNT\System32\MSRLE32.DLL
        5.00.2134.1   10.77 KB (11,024 bytes)   10/31/2002 2:23:20
AM
c:\winnt\system32\msh263.drv   Microsoft Corporation
        OK   C:\WINNT\System32\MSH263.DRV   4.4.3385
        252.27 KB (258,320 bytes)   10/31/2002 10:46:31 AM
c:\winnt\system32\iccvid.dll   Radius Inc.   OK
        C:\WINNT\System32\ICCVID.DLL   1.10.0.6   108.00
KB (110,592 bytes)   10/31/2002 2:21:10 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)   10/31/2002 2:21:46 AM

[CD-ROM]

Item   Value
Drive   D:
Description   CD-ROM Drive
Media Loaded   False
Media Type   CD-ROM
Name   COMPAQ CD-224E
Manufacturer   (Standard CD-ROM drives)
Status   OK
Transfer Rate   Not Available
SCSI Target ID   0
PNP Device ID   IDE\CDROMCOMPAQ_CD-
224E_____A.8D____\5&23A72C42&0&0.0.0


[Sound Device]

Item   Value
No sound devices

[Display]

Item   Value
Name   ATI Technologies Inc. RAGE XL PCI
```

PNP Device ID
PCI\VEN_1002&DEV_4752&SUBSYS_001E0E11&REV_27\3&267A616A&0&18
Adapter Type ATI RAGE XL PCI, ATI Technologies Inc. compatible
Adapter Description ATI Technologies Inc. RAGE XL PCI
Adapter RAM 8.00 MB (8,388,608 bytes)
Installed Drivers atidrab.dll
Driver Version 5.00.2179.1
INF File display.inf (atirage3 section)
Color Planes 1
Color Table Entries 256
Resolution 1024 x 768 x 60 hertz
Bits/Pixel 8

[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&32BA4B66&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&32BA4B66&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] Compaq NC7780 Gigabit Server Adapter
Adapter Type Ethernet 802.3
Product Name Compaq NC7780 Gigabit Server Adapter
Installed True
PNP Device ID
PCI\VEN_14E4&DEV_1645&SUBSYS_00850E11&REV_15\3&13C0B0C5&0&28
Last Reset 12/4/2002 9:09:15 AM
Index 0
Service Name q57w2k
IP Address 15.1.100.41
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled True
DHCP Server 15.1.100.1
DHCP Lease Expires 12/16/2002 5:09:52 PM
DHCP Lease Obtained 12/8/2002 5:09:52 PM
MAC Address 00:08:02:A3:D4:60
Service Name q57w2k
IRQ Number 30
Driver c:\winnt\system32\drivers\q57w2k.sys (77776, 2.75.0.0)

Name [00000001] Compaq NC7780 Gigabit Server Adapter
Adapter Type Ethernet 802.3
Product Name Compaq NC7780 Gigabit Server Adapter
Installed True
PNP Device ID
PCI\VEN_14E4&DEV_1645&SUBSYS_00850E11&REV_15\3&13C0B0C5&0&30
Last Reset 12/4/2002 9:09:15 AM
Index 1
Service Name q57w2k
IP Address
IP Subnet
Default IP Gateway Not Available
DHCP Enabled True
DHCP Server
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:08:02:A3:D4:6B
Service Name q57w2k
IRQ Number 29
Driver c:\winnt\system32\drivers\q57w2k.sys (77776, 2.75.0.0)

Name [00000002] RAS Async Adapter
Adapter Type Not Available
Product Name RAS Async Adapter
Installed True
PNP Device ID Not Available
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 12/4/2002 9:09:15 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 (52112, 5.00.2195.4052)

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 12/4/2002 9:09:15 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 (47888, 5.00.2195.4080)

Name [00000005] Direct Parallel
Adapter Type Not Available

Product Name Direct Parallel
Installed True
PNP Device ID ROOT\MS_PTIMINIPORT\0000
Last Reset 12/4/2002 9:09:15 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 12/4/2002 9:09:15 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 (93104, 5.00.2195.5241)

[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 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 False
SupportsExpeditedData True
SupportsGracefulClosing True
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{1884A550-A4C0-4616-8C0E-9FC8862D2F3A}] 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_{1884A550-A4C0-4616-8C0E-9FC8862D2F3A}] 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_{D7733D23-53A5-48A7-8864-FB1E0C26BF6D}] 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_{D7733D23-53A5-48A7-8864-FB1E0C26BF6D}] 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_{2CEAB79C-DF80-42D8-9B6B-3C7E20CABC5D}] 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_{2CEAB79C-DF80-42D8-9B6B-3C7E20CABC5D}] DATAGRAM 2
ConnectionlessService True
GuaranteesDelivery False
MaximumAddressSize 20 bytes
GuaranteesSequencing False
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_{A4470085-E1C1-456F-9466-91EE411F43E5}] 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_{A4470085-E1C1-456F-9466-91EE411F43E5}] 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 5.00.2195.4874
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\0
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 (62512, 5.00.2195.5080)

[Parallel]

Item Value
No parallel port information

[Storage]

[Following are sub-categories of this main category]

[Drives]

Item Value
Drive A:
Description 3 1/2 Inch Floppy Drive

Drive C:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 16.91 GB (18,157,239,808 bytes)
Free Space 14.03 GB (15,063,959,552 bytes)
Volume Name
Volume Serial Number B8528649
Partition Disk #0, Partition #1
Partition Size 16.91 GB (18,157,240,320 bytes)
Starting Offset 37601280 bytes
Drive Description Disk drive
Drive Manufacturer (Standard disk drives)
Drive Model COMPAQ LOGICAL VOLUME SCSI Disk Device
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 2
Drive SCSITargetId 4
Drive SectorsPerTrack 32
Drive Size 18203197440 bytes
Drive TotalCylinders 4357
Drive TotalSectors 35553120
Drive TotalTracks 1111035
Drive TracksPerCylinder 255

[SCSI]

Item Value
Name Compaq Smart Array 5i
Caption Compaq Smart Array 5i
Driver cpqcissm
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B178&SUBSYS_40800E11&REV_01\3&13C0B0C5&0&20
Device ID
PCI\VEN_0E11&DEV_B178&SUBSYS_40800E11&REV_01\3&13C0B0C5&0&20
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 31
I/O Port 0x3000-0x30FF
Driver c:\winnt\system32\drivers\cpqcissm.sys (14992, 5.40.2.0)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Caption QLogic QLA23xx PCI Fibre Channel Adapter
Driver ql2300
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&1070020&0&30
Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&1070020&0&30
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 26
I/O Port 0x4000-0x40FF
Driver c:\winnt\system32\drivers\ql2300.sys (440012, 8.2.0 Beta 3 (W2K VI))

[Printing]

Name Port Name Server Name
No printing information

[Problem Devices]

Device PNP Device ID Error Code
No Problem Devices

[USB]

Device PNP Device ID
Standard OpenHCD USB Host Controller
PCI\VEN_1166&DEV_0220&SUBSYS_02201166&REV_05\3&267A616A&0&7A
USB Root Hub USB\ROOT_HUB\4&AF5358C&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
atirage3       atirage3  c:\winnt\system32\drivers\atimpab.sys  Kernel
Driver         True      Manual    Running   OK        Ignore
               False     True
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
cnmprot        Compaq Network Management Protocol Driver
               c:\winnt\system32\drivers\cnmprot.sys    Kernel Driver False
               Manual    Stopped   OK        Normal    False
               False
cpq32fs2       cpq32fs2  c:\winnt\system32\drivers\cpq32fs2.sys   Kernel
Driver         True      Boot      Running   OK        Normal
               False     True
cpqarray       Cpqarray  c:\winnt\system32\drivers\cpqarray.sys   Kernel
Driver         True      Boot      Running   OK        Normal
               False     True
cpqarry2       cpqarry2  c:\winnt\system32\drivers\cpqarry2.sys   Kernel
Driver         True      Boot      Running   OK        Normal
               False     True
cpqasm2        Compaq iLO Advanced System Management Controller
               c:\winnt\system32\drivers\cpqasm2.sys    Kernel Driver True
               Manual    Running   OK        Normal    False
               True
cpqcidrv       Compaq iLO Management Interface Driver
               c:\winnt\system32\drivers\cpqcidrv.sys   Kernel Driver True
               Manual    Running   OK        Normal    False
               True
cpqcisse       CPQCISSE  c:\winnt\system32\drivers\cpqcisse.sys   Kernel
Driver         True      System    Running   OK        Normal
               False     True
cpqcissm       cpqcissm  c:\winnt\system32\drivers\cpqcissm.sys   Kernel
Driver         True      Boot      Running   OK        Normal
               False     True
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
cpqteam        Compaq Network Teaming and Configuration
               c:\winnt\system32\drivers\cpqteam.sys    Kernel Driver False
               Manual    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
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
fips           Fips      c:\winnt\system32\drivers\fips.sys       Kernel
Driver         True      Auto      Running   OK        Normal
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
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
irenum         IR Enumerator Service
               c:\winnt\system32\drivers\irenum.sys     Kernel Driver False
               Manual    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
megaide         MegaIDE         c:\winnt\system32\drivers\megaide.sys   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        c:\winnt\system32\drivers\parallel.sys  Kernel
Driver          False           Manual          Stopped         OK              Ignore
                False           False
parport         Parport         c:\winnt\system32\drivers\parport.sys   Kernel
Driver          False           Manual          Stopped         OK              Ignore
                False           False
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          False           Manual          Stopped         OK              Ignore
pci             PCI Bus Driver  False           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
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
q57w2k          Compaq NC7780 Gigabit Server Adapter
                c:\winnt\system32\drivers\q57w2k.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
ql2300          ql2300          c:\winnt\system32\drivers\ql2300.sys    Kernel
Driver          True            Boot            Running         OK              Normal
                False           True
qlvika          qlvika          c:\winnt\system32\drivers\qlvika.sys    Kernel
Driver          True            Auto            Running         OK              Normal
                False           True
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
rdpdr           Terminal Server Device Redirector Driver
                c:\winnt\system32\drivers\rdpdr.sys      Kernel Driver True
                Manual  Running OK      Normal  False
                True
rdpwd           RDPWD   c:\winnt\system32\drivers\rdpwd.sys      Kernel
Driver          True    Manual  Running OK      Ignore
                False   True
redbook         Digital CD Audio Playback Filter Driver
                c:\winnt\system32\drivers\redbook.sys    Kernel Driver False
                System  Stopped OK      Normal  False
                False
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      Manual  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 c:\winnt\system32\drivers\symc810.sys    Kernel
Driver          True    Boot    Running OK      Normal
symc8xx         symc8xx c:\winnt\system32\drivers\symc8xx.sys    Kernel
Driver          True    Boot    Running OK      Normal
                False   True
sym_hi          sym_hi  c:\winnt\system32\drivers\sym_hi.sys    Kernel
Driver          True    Boot    Running OK      Normal
                False   True
sysmgmt         Compaq System Management Interface Driver
                c:\winnt\system32\drivers\sysmgmt.sys    Kernel Driver False
                Manual  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          True    Manual  Running OK      Ignore
                False   True
termdd          Terminal Device Driver
                c:\winnt\system32\drivers\termdd.sys     Kernel Driver True
                Auto    Running OK      Normal  False
                True
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
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%\Sy
stem32\Wbem;C:\Program Files\Microsoft SQL Server\80\Tools\BINN
                <SYSTEM>
windir          %SystemRoot%            <SYSTEM>
OS              Windows_NT <SYSTEM>
PROCESSOR_ARCHITECTURE          x86     <SYSTEM>
PROCESSOR_LEVEL 6               <SYSTEM>
PROCESSOR_IDENTIFIER  x86 Family 6 Model 11 Stepping 4, GenuineIntel
                <SYSTEM>
PROCESSOR_REVISION      0b04    <SYSTEM>
NUMBER_OF_PROCESSORS            2       <SYSTEM>
PATHEXT         .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
                <SYSTEM>
TEMP            %SystemRoot%\TEMP       <SYSTEM>
TMP             %SystemRoot%\TEMP       <SYSTEM>
TEMP            %USERPROFILE%\Local Settings\Temp
                TPCWEB1\Administrator
TMP             %USERPROFILE%\Local Settings\Temp
                TPCWEB1\Administrator

[Jobs]

[ Following are sub-categories of this main category ]

[Print]

Document        Size    Owner   Notify  Status  Time
Submitted       Start Time      Until Time      Elapsed Time    Pages Printed
                Job ID  Priority        Parameters      Driver Name     Print
Processor       Host Print Queue        Data Type       Name
No print jobs




[Network Connections]
```

Local Name	Remote Name	Type	Status	User Name
E:	\\ssdllab\d$	Disk	OK	

[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	224	11	204800	1413120	12/4/2002 5:09:46 PM	5.00.2195.5382	44.77 KB (45,840 bytes)	10/31/2002 2:25:47 AM
csrss.exe	Not Available	252	13	Not Available	Not Available	12/4/2002 5:09:49 PM	Unknown	Unknown	Unknown
winlogon.exe	c:\winnt\system32\winlogon.exe	248	13	204800	1413120	12/4/2002 5:09:50 PM	5.00.2195.5386	174.77 KB (178,960 bytes)	6/25/2002 4:07:22 AM
services.exe	c:\winnt\system32\services.exe	300	9	204800	1413120	12/4/2002 5:09:50 PM	5.00.2195.3940	86.77 KB (88,848 bytes)	10/31/2002 2:25:27 AM
lsass.exe	c:\winnt\system32\lsass.exe	312	9	204800	1413120	12/4/2002 5:09:50 PM	5.00.2195.5430	32.77 KB (33,552 bytes)	10/31/2002 2:22:13 AM
termsrv.exe	c:\winnt\system32\termsrv.exe	420	10	204800	1413120	12/4/2002 5:09:51 PM	5.00.2195.5276	138.77 KB (142,096 bytes)	6/25/2002 4:07:21 AM
svchost.exe	c:\winnt\system32\svchost.exe	520	8	204800	1413120	12/4/2002 5:09:52 PM	5.00.2134.1	7.77 KB (7,952 bytes)	10/31/2002 2:26:11 AM
spoolsv.exe	c:\winnt\system32\spoolsv.exe	556	8	204800	1413120	12/4/2002 5:09:52 PM	5.00.2195.4299	44.27 KB (45,328 bytes)	10/31/2002 2:38:21 AM
msdtc.exe	c:\winnt\system32\msdtc.exe	584	8	204800	1413120	12/4/2002 5:09:52 PM	1999.9.3421.3	6.77 KB (6,928 bytes)	10/31/2002 2:44:23 AM
svchost.exe	c:\winnt\system32\svchost.exe	708	8	204800	1413120	12/4/2002 5:09:53 PM	5.00.2134.1	7.77 KB (7,952 bytes)	10/31/2002 2:26:11 AM
llssrv.exe	c:\winnt\system32\llssrv.exe	728	9	204800	1413120	12/4/2002 5:09:53 PM	5.00.2195.4907	81.27 KB (83,216 bytes)	7/22/2002 12:05:04 PM
regsvc.exe	c:\winnt\system32\regsvc.exe	776	8	204800	1413120	12/4/2002 5:09:54 PM	5.00.2195.3649	65.27 KB (66,832 bytes)	6/25/2002 4:07:17 AM
rsys.exe	Not Available	828	8	Not Available	Not Available	12/4/2002 5:09:54 PM	Unknown	Unknown	Unknown
mstask.exe	c:\winnt\system32\mstask.exe	840	8	204800	1413120	12/4/2002 5:09:54 PM	4.71.2195.1	115.77 KB (118,544 bytes)	6/25/2002 4:07:12 AM
snmp.exe	c:\winnt\system32\snmp.exe	876	8	204800	1413120	12/4/2002 5:09:54 PM	5.00.2195.5080	29.77 KB (30,480 bytes)	10/31/2002 2:44:21 AM
surveyor.exe	c:\compaq\survey\surveyor.exe	916	8	204800	1413120	12/4/2002 5:09:54 PM	2.44	1015.80 KB (1,040,176 bytes)	6/25/2002 3:32:45 AM
winmgmt.exe	c:\winnt\system32\wbem\winmgmt.exe	960	8	204800	1413120	12/4/2002 5:09:55 PM	1.50.1085.0070	192.08 KB (196,685 bytes)	6/25/2002 4:07:27 AM
svchost.exe	c:\winnt\system32\svchost.exe	972	8	204800	1413120	12/4/2002 5:09:55 PM	5.00.2134.1	7.77 KB (7,952 bytes)	10/31/2002 2:26:11 AM
dfssvc.exe	c:\winnt\system32\dfssvc.exe	996	8	204800	1413120	12/4/2002 5:09:55 PM	5.00.2195.3649	88.27 KB (90,384 bytes)	6/25/2002 4:07:02 AM
svchost.exe	c:\winnt\system32\svchost.exe	1384	8	204800	1413120	12/4/2002 5:10:24 PM	5.00.2134.1	7.77 KB (7,952 bytes)	10/31/2002 2:26:11 AM
dllhost.exe	Not Available	732	8	Not Available	Not Available	12/5/2002 11:21:21 AM	Unknown	Unknown	Unknown
csrss.exe	Not Available	9212	13	Not Available	Not Available	12/6/2002 9:20:53 PM	Unknown	Unknown	Unknown
winlogon.exe	c:\winnt\system32\winlogon.exe	9320	13	204800	1413120	12/6/2002 9:20:53 PM	5.00.2195.5386	174.77 KB (178,960 bytes)	6/25/2002 4:07:22 AM
rdpclip.exe	c:\winnt\system32\rdpclip.exe	9268	8	204800	1413120	12/6/2002 9:20:57 PM	5.00.2174.1	39.77 KB (40,720 bytes)	10/31/2002 2:44:23 AM
explorer.exe	c:\winnt\explorer.exe	8524	8	204800	1413120	12/6/2002 9:20:57 PM	5.00.3502.5321	237.27 KB (242,960 bytes)	6/25/2002 4:07:23 AM
logon.scr	c:\winnt\system32\logon.scr	9144	4	204800	1413120	12/8/2002 5:46:51 PM	5.00.2195.5305	127.77 KB (130,832 bytes)	6/25/2002 4:07:06 AM
inetinfo.exe	c:\winnt\system32\inetsrv\inetinfo.exe	7684	8	204800	1413120	12/9/2002 2:45:42 PM	5.00.0984	14.27 KB (14,608 bytes)	6/25/2002 4:07:37 AM
mmc.exe	c:\winnt\system32\mmc.exe	3812	8	204800	1413120	12/9/2002 3:58:39 PM	5.00.2195.4933	589.27 KB (603,408 bytes)	6/25/2002 4:07:08 AM
rsvp.exe	c:\winnt\system32\rsvp.exe	3744	8	204800	1413120	12/9/2002 4:00:55 PM	5.00.2167.1	172.77 KB (176,912 bytes)	10/31/2002 2:25:12 AM

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
traffic.dll	5.00.2139.1	30.77 KB (31,504 bytes)	10/31/2002 2:26:30 AM	Microsoft Corporation	c:\winnt\system32\traffic.dll
rsvp.exe	5.00.2167.1	172.77 KB (176,912 bytes)	10/31/2002 2:25:12 AM	Microsoft Corporation	c:\winnt\system32\rsvp.exe
wbemprox.dll	1.50.1085.0045	40.08 KB (41,040 bytes)	6/25/2002 4:07:27 AM	Microsoft Corporation	c:\winnt\system32\wbem\wbemprox.dll
mlang.dll	5.00.3315.3727	509.77 KB (522,000 bytes)	6/25/2002 4:07:08 AM	Microsoft Corporation	c:\winnt\system32\mlang.dll
cabinet.dll	5.00.2147.1	54.77 KB (56,080 bytes)	10/31/2002 2:18:17 AM	Microsoft Corporation	c:\winnt\system32\cabinet.dll
msinfo32.dll	5.00.2195.4601	312.27 KB (319,760 bytes)	6/25/2002 4:07:28 AM	Microsoft Corporation	c:\program files\common files\microsoft shared\msinfo\msinfo32.dll
mmcndmgr.dll	5.00.2195.5352	816.27 KB (835,856 bytes)	6/25/2002 4:07:08 AM	Microsoft Corporation	c:\winnt\system32\mmcndmgr.dll
mmc.exe	5.00.2195.4933	589.27 KB (603,408 bytes)	6/25/2002 4:07:08 AM	Microsoft Corporation	c:\winnt\system32\mmc.exe
httpext.dll	5.00.0984	240.27 KB (246,032 bytes)	6/25/2002 4:07:37 AM	Microsoft Corporation	c:\winnt\system32\inetsrv\httpext.dll
iscomlog.dll	5.00.0984	24.27 KB (24,848 bytes)	6/25/2002 4:07:37 AM	Microsoft Corporation	c:\winnt\system32\inetsrv\iscomlog.dll
lonsint.dll	5.00.0984	11.77 KB (12,048 bytes)	6/25/2002 4:07:37 AM	Microsoft Corporation	c:\winnt\system32\inetsrv\lonsint.dll
inetsloc.dll	5.00.0984	20.27 KB (20,752 bytes)	6/25/2002 4:07:06 AM	Microsoft Corporation	c:\winnt\system32\inetsloc.dll
iisfecnv.dll	5.00.0984	7.27 KB (7,440 bytes)	10/31/2002 2:44:31 AM	Microsoft Corporation	c:\winnt\system32\inetsrv\iisfecnv.dll

isatq.dll 5.00.0984 60.77 KB (62,224 bytes) 6/25/2002 4:07:37 AM Microsoft Corporation c:\winnt\system32\inetsrv\isatq.dll
infocomm.dll 5.00.0984 240.77 KB (246,544 bytes) 6/25/2002 4:07:37 AM Microsoft Corporation c:\winnt\system32\inetsrv\infocomm.dll
w3svc.dll 5.00.0984 335.27 KB (343,312 bytes) 6/25/2002 4:07:38 AM Microsoft Corporation c:\winnt\system32\inetsrv\w3svc.dll
security.dll 5.00.2154.1 5.77 KB (5,904 bytes) 10/31/2002 2:25:25 AM Microsoft Corporation c:\winnt\system32\security.dll
svcext.dll 5.00.0984 39.77 KB (40,720 bytes) 6/25/2002 4:07:38 AM Microsoft Corporation c:\winnt\system32\inetsrv\svcext.dll
admexs.dll 5.00.0984 27.77 KB (28,432 bytes) 6/25/2002 4:07:36 AM Microsoft Corporation c:\winnt\system32\inetsrv\admexs.dll
wamreg.dll 5.00.0984 45.77 KB (46,864 bytes) 6/25/2002 4:07:38 AM Microsoft Corporation c:\winnt\system32\inetsrv\wamreg.dll
metadata.dll 5.00.0984 68.77 KB (70,416 bytes) 6/25/2002 4:07:37 AM Microsoft Corporation c:\winnt\system32\inetsrv\metadata.dll
iismap.dll 5.00.0984 55.77 KB (57,104 bytes) 6/25/2002 4:07:06 AM Microsoft Corporation c:\winnt\system32\iismap.dll
nsepm.dll 5.00.0984 43.27 KB (44,304 bytes) 6/25/2002 4:07:37 AM Microsoft Corporation c:\winnt\system32\inetsrv\nsepm.dll
admwprox.dll 5.00.0984 31.77 KB (32,528 bytes) 10/31/2002 2:44:31 AM Microsoft Corporation c:\winnt\system32\admwprox.dll
coadmin.dll 5.00.0984 39.77 KB (40,720 bytes) 6/25/2002 4:07:37 AM Microsoft Corporation c:\winnt\system32\inetsrv\coadmin.dll
iisadmin.dll 5.00.0984 15.27 KB (15,632 bytes) 6/25/2002 4:07:37 AM Microsoft Corporation c:\winnt\system32\inetsrv\iisadmin.dll
rpcref.dll 5.00.0984 4.27 KB (4,368 bytes) 6/25/2002 4:07:38 AM Microsoft Corporation c:\winnt\system32\inetsrv\rpcref.dll
inetinfo.exe 5.00.0984 14.27 KB (14,608 bytes) 6/25/2002 4:07:37 AM Microsoft Corporation c:\winnt\system32\inetsrv\inetinfo.exe
logon.scr 5.00.2195.5305 127.77 KB (130,832 bytes) 6/25/2002 4:07:08 AM Microsoft Corporation c:\winnt\system32\logon.scr
shdoclc.dll 5.00.3502.5039 324.50 KB (332,288 bytes) 6/25/2002 4:07:19 AM Microsoft Corporation c:\winnt\system32\shdoclc.dll
msi.dll 2.0.2600.1 1.90 MB (1,991,168 bytes) 6/25/2002 4:07:10 AM Microsoft Corporation c:\winnt\system32\msi.dll
urlmon.dll 5.00.3502.5400 442.27 KB (452,880 bytes) 6/25/2002 4:07:22 AM Microsoft Corporation c:\winnt\system32\urlmon.dll
faxshell.dll 5.00.2134.1 8.27 KB (8,464 bytes) 10/31/2002 2:20:39 AM Microsoft Corporation c:\winnt\system32\faxshell.dll
msacm32.dll 5.00.2134.1 65.27 KB (66,832 bytes) 10/31/2002 2:22:53 AM Microsoft Corporation c:\winnt\system32\msacm32.dll
avifil32.dll 5.00.2134.1 76.27 KB (78,096 bytes) 10/31/2002 2:18:03 AM Microsoft Corporation c:\winnt\system32\avifil32.dll
msvfw32.dll 5.00.2134.1 13.77 KB (14,096 bytes) 10/31/2002 2:03:27 AM Microsoft Corporation c:\winnt\system32\msvfw32.dll
msrhui.dll 5.00.2134.1 113.77 KB (116,496 bytes) 10/31/2002 2:03:27 AM Microsoft Corporation c:\winnt\system32\msrhui.dll
docprop2.dll 5.00.2178.1 297.77 KB (304,912 bytes) 10/31/2002 2:19:01 AM Microsoft Corporation c:\winnt\system32\docprop2.dll
linkinfo.dll 5.00.2134.1 15.77 KB (16,144 bytes) 10/31/2002 2:22:07 AM Microsoft Corporation c:\winnt\system32\linkinfo.dll
powrprof.dll 5.00.3502.5305 13.27 KB (13,584 bytes) 6/25/2002 4:07:17 AM Microsoft Corporation c:\winnt\system32\powrprof.dll
batmeter.dll 5.00.3502.5305 20.27 KB (20,752 bytes) 6/25/2002 4:06:58 AM Microsoft Corporation c:\winnt\system32\batmeter.dll
stobject.dll 5.00.2195.4455 79.27 KB (81,168 bytes) 6/25/2002 4:07:20 AM Microsoft Corporation c:\winnt\system32\stobject.dll
webcheck.dll 5.00.3315.3727 250.77 KB (256,784 bytes) 6/25/2002 4:07:22 AM Microsoft Corporation c:\winnt\system32\webcheck.dll
browselc.dll 5.00.3502.4373 34.50 KB (35,328 bytes) 6/25/2002 4:06:58 AM Microsoft Corporation c:\winnt\system32\browselc.dll
ntshrui.dll 5.00.2134.1 46.77 KB (47,888 bytes) 10/31/2002 2:24:12 AM Microsoft Corporation c:\winnt\system32\ntshrui.dll
mydocs.dll 5.00.3315.4065 55.27 KB (56,592 bytes) 6/25/2002 4:07:13 AM Microsoft Corporation c:\winnt\system32\mydocs.dll
browseui.dll 5.00.3502.4373 791.27 KB (810,256 bytes) 6/25/2002 4:06:58 AM Microsoft Corporation c:\winnt\system32\browseui.dll
shdocvw.dll 5.00.3502.5400 1.05 MB (1,105,168 bytes) 6/25/2002 4:07:19 AM Microsoft Corporation c:\winnt\system32\shdocvw.dll
explorer.exe 5.00.3502.5321 237.27 KB (242,960 bytes) 6/25/2002 4:07:23 AM Microsoft Corporation c:\winnt\explorer.exe
rdpclip.exe 5.00.2174.1 39.77 KB (40,720 bytes) 10/31/2002 2:44:23 AM Microsoft Corporation c:\winnt\system32\rdpclip.exe
mscms.dll 5.00.2180.1 68.27 KB (69,904 bytes) 10/31/2002 2:22:57 AM Microsoft Corporation c:\winnt\system32\mscms.dll
printui.dll 5.00.2195.5212 372.27 KB (381,200 bytes) 10/31/2002 2:24:40 AM Microsoft Corporation c:\winnt\system32\printui.dll
netmsg.dll 5.00.2137.1 152.50 KB (156,160 bytes) 10/31/2002 2:23:40 AM Microsoft Corporation c:\winnt\system32\netmsg.dll
netui2.dll 5.00.2134.1 280.27 KB (286,992 bytes) 10/31/2002 2:23:43 AM Microsoft Corporation c:\winnt\system32\netui2.dll
mprui.dll 5.00.2195.4874 54.77 KB (56,080 bytes) 6/25/2002 4:07:09 AM Microsoft Corporation c:\winnt\system32\mprui.dll
cscui.dll 5.00.2195.4104 233.77 KB (239,376 bytes) 6/25/2002 4:07:01 AM Microsoft Corporation c:\winnt\system32\cscui.dll
tapisrv.dll 5.00.2195.5227 169.27 KB (173,328 bytes) 6/25/2002 4:07:21 AM Microsoft Corporation c:\winnt\system32\tapisrv.dll
dfssvc.exe 5.00.2195.3649 88.27 KB (90,384 bytes) 6/25/2002 4:07:02 AM Microsoft Corporation c:\winnt\system32\dfssvc.exe
sensapi.dll 5.00.2163.1 6.77 KB (6,928 bytes) 10/31/2002 2:25:25 AM Microsoft Corporation c:\winnt\system32\sensapi.dll
winhttp.dll 5.1.2600.1039 (xpsp1.020511-1800) 303.00 KB (310,272 bytes) 6/25/2002 4:07:36 AM Microsoft Corporation c:\winnt\system32\winhttp.dll
wininet.dll 5.00.3502.4619 450.77 KB (461,584 bytes) 6/25/2002 4:07:22 AM Microsoft Corporation c:\winnt\system32\wininet.dll
utildll.dll 5.00.2153.1 25.77 KB (26,384 bytes) 10/31/2002 2:27:02 AM Microsoft Corporation c:\winnt\system32\utildll.dll
wtsapi32.dll 5.00.2134.1 14.27 KB (14,608 bytes) 10/31/2002 2:28:05 AM Microsoft Corporation c:\winnt\system32\wtsapi32.dll
advpack.dll 5.00.3502.4373 86.77 KB (88,848 bytes) 6/25/2002 4:06:58 AM Microsoft Corporation c:\winnt\system32\advpack.dll
wuaueng.dll 5.4.3628.1 built by: lab04_n 182.50 KB (186,880 bytes) 6/25/2002 4:07:36 AM Microsoft Corporation c:\winnt\system32\wuaueng.dll
wuauserv.dll 5.4.3628.1 built by: lab04_n 8.50 KB (8,704 bytes) 6/25/2002 4:07:36 AM Microsoft Corporation c:\winnt\system32\wuauserv.dll
netui1.dll 5.00.2134.1 210.27 KB (215,312 bytes) 10/31/2002 2:23:43 AM Microsoft Corporation c:\winnt\system32\netui1.dll
netui0.dll 5.00.2195.4874 70.77 KB (72,464 bytes) 6/25/2002 4:07:14 AM Microsoft Corporation c:\winnt\system32\netui0.dll
ntlanman.dll 5.00.2195.5428 35.27 KB (36,112 bytes) 10/31/2002 2:24:04 AM Microsoft Corporation c:\winnt\system32\ntlanman.dll
wshnetbs.dll 5.00.2134.1 7.77 KB (7,952 bytes) 10/31/2002 2:28:05 AM Microsoft Corporation c:\winnt\system32\wshnetbs.dll
ntmarta.dll 5.00.2195.4836 99.77 KB (102,160 bytes) 6/25/2002 4:07:15 AM Microsoft Corporation c:\winnt\system32\ntmarta.dll
provthrd.dll 1.50.1085.0000 68.07 KB (69,708 bytes) 10/31/2002 10:46:37 AM Microsoft Corporation c:\winnt\system32\wbem\provthrd.dll
ntevt.dll 1.50.1085.0072 192.06 KB (196,671 bytes) 6/25/2002 4:07:26 AM Microsoft Corporation c:\winnt\system32\wbem\ntevt.dll
psapi.dll 5.00.2134.1 28.27 KB (28,944 bytes) 10/31/2002 2:24:43 AM Microsoft Corporation c:\winnt\system32\psapi.dll
framedyn.dll 1.50.1085.0076 164.07 KB (168,009 bytes) 6/25/2002 4:07:26 AM Microsoft Corporation c:\winnt\system32\wbem\framedyn.dll
cimwin32.dll 1.50.1085.0073 1.04 MB (1,085,520 bytes) 6/25/2002 4:07:26 AM Microsoft Corporation c:\winnt\system32\wbem\cimwin32.dll

wbemsvc.dll 1.50.1085.0007 40.07 KB (41,036 bytes)
6/25/2002 4:07:27 AM Microsoft Corporation
c:\winnt\system32\wbem\wbemsvc.dll
wbemess.dll 1.50.1085.0074 364.07 KB (372,804 bytes)
6/25/2002 4:07:27 AM Microsoft Corporation
c:\winnt\system32\wbem\wbemess.dll
fastprox.dll 1.50.1085.0056 144.08 KB (147,536 bytes)
6/25/2002 4:07:26 AM Microsoft Corporation
c:\winnt\system32\wbem\fastprox.dll
wbemcore.dll 1.50.1085.0085 628.07 KB (643,146 bytes)
6/25/2002 4:07:26 AM Microsoft Corporation
c:\winnt\system32\wbem\wbemcore.dll
wbemcomn.dll 1.50.1085.0077 692.07 KB (708,675 bytes) 6/25/2002 4:07:26 AM Microsoft Corporation
c:\winnt\system32\wbem\wbemcomn.dll
winmgmt.exe 1.50.1085.0070 192.08 KB (196,685 bytes)
6/25/2002 4:07:27 AM Microsoft Corporation
c:\winnt\system32\wbem\winmgmt.exe
mssngrus.dll 2.44 164.07 KB (168,008 bytes) 6/25/2002 3:32:45 AM
Compaq Computer Corp. c:\compaq\survey\mssngrus.dll
comdlg32.dll 5.00.3315.3727 221.27 KB (226,576 bytes)
10/31/2002 2:18:35 AM Microsoft Corporation
c:\winnt\system32\comdlg32.dll
expat.dll Not Available 144.00 KB (147,456 bytes) 6/25/2002 3:32:45 AM
Not Available c:\compaq\survey\expat.dll
cpqhmmo.dll 5.0.0 987.19 KB (1,010,880 bytes)
6/25/2002 3:32:45 AM Compaq Computer Corp.
c:\compaq\survey\cpqhmmo.dll
surveyor.exe 2.44 1015.80 KB (1,040,176 bytes)
6/25/2002 3:32:45 AM Compaq Computer Corp.
c:\compaq\survey\surveyor.exe
sm2user.dll 5.50.0.0 152.07 KB (155,722 bytes) 6/25/2002 3:31:04 AM
Compaq Computer Corp. c:\winnt\system32\sm2user.dll
cpqnimib.dll Not Available 31.38 KB (32,128 bytes) 6/25/2002 3:32:14 AM
Not Available c:\winnt\system32\cpqnimgt\cpqnimib.dll
cqnisnmp.dll Not Available 7.45 KB (7,632 bytes) 6/25/2002 3:32:14 AM
Not Available c:\winnt\system32\cpqnimgt\cqnisnmp.dll
cpqniutl.dll Not Available 16.19 KB (16,576 bytes) 6/25/2002 3:32:14 AM
Not Available c:\winnt\system32\cpqniutl.dll
nicmib.dll Not Available 14.92 KB (15,280 bytes) 6/25/2002 3:32:14 AM
Not Available c:\winnt\system32\cpqnimgt\nicmib.dll
storsnmp.dll Not Available 7.45 KB (7,632 bytes) 6/25/2002 3:31:38 AM
Not Available
c:\winnt\system32\cpqmgmt\cqmgstor\storsnmp.dll
cqstrutl.dll Not Available 28.64 KB (29,328 bytes) 6/25/2002 3:31:38 AM
Not Available c:\winnt\system32\cqstrutl.dll
stormib.dll Not Available 183.89 KB (188,304 bytes) 6/25/2002 3:31:38 AM
Not Available c:\winnt\system32\cpqmgmt\cqmgstor\stormib.dll
servsnmp.dll Not Available 7.45 KB (7,632 bytes) 6/25/2002 3:31:04 AM
Not Available
c:\winnt\system32\cpqmgmt\cqmgserv\servsnmp.dll

cqsrvutl.dll Not Available 31.30 KB (32,048 bytes) 6/25/2002 3:31:04 AM
servmib.dll Not Available 186.38 KB (190,848 bytes) 6/25/2002 3:31:04 AM
Not Available
c:\winnt\system32\cpqmgmt\cqmgserv\servmib.dll
cpqmib1k.dll Not Available 3.31 KB (3,392 bytes) 6/25/2002 3:30:31 AM
Not Available
c:\winnt\system32\cpqmgmt\cqmghost\cpqmib1k.dll
hostsnmp.dll Not Available 9.83 KB (10,064 bytes) 6/25/2002 3:30:31 AM
Not Available
c:\winnt\system32\cpqmgmt\cqmghost\hostsnmp.dll
cqhstutl.dll Not Available 30.69 KB (31,424 bytes) 6/25/2002 3:30:31 AM
Not Available c:\winnt\system32\cqhstutl.dll
crtdll.dll 4.00 145.77 KB (149,264 bytes) 10/31/2002 2:18:42 AM
Microsoft Corporation c:\winnt\system32\crtdll.dll
hostmib.dll Not Available 110.88 KB (113,536 bytes) 6/25/2002 3:30:31 AM
Not Available
c:\winnt\system32\cpqmgmt\cqmghost\hostmib.dll
iisrtl.dll 5.00.0984 119.77 KB (122,640 bytes) 6/25/2002 4:07:06 AM
Microsoft Corporation c:\winnt\system32\iisrtl.dll
infoadmn.dll 5.00.0984 13.27 KB (13,584 bytes) 6/25/2002 4:07:06 AM
Microsoft Corporation c:\winnt\system32\infoadmn.dll
httpmib.dll 5.00.0984 9.27 KB (9,488 bytes) 6/25/2002 4:07:37 AM
Microsoft Corporation
c:\winnt\system32\inetsrv\httpmib.dll
rtipxmib.dll 5.00.2168.1 29.77 KB (30,480 bytes) 10/31/2002 2:25:12 AM
Microsoft Corporation c:\winnt\system32\rtipxmib.dll
btpagnt.dll 5.00.2168.1 13.27 KB (13,584 bytes) 10/31/2002 2:18:10 AM
Microsoft Corporation c:\winnt\system32\btpagnt.dll
ospfagnt.dll 5.00.2168.1 6.77 KB (6,928 bytes) 10/31/2002 2:24:26 AM
Microsoft Corporation c:\winnt\system32\ospfagnt.dll
ripagnt.dll 5.00.2168.1 24.27 KB (24,848 bytes) 10/31/2002 2:25:03 AM
Microsoft Corporation c:\winnt\system32\ripagnt.dll
perfos.dll 5.00.2155.1 21.27 KB (21,776 bytes) 10/31/2002 2:24:34 AM
Microsoft Corporation c:\winnt\system32\perfos.dll
mcastmib.dll 5.00.2168.1 13.27 KB (13,584 bytes) 10/31/2002 2:22:15 AM
Microsoft Corporation c:\winnt\system32\mcastmib.dll
igmpagnt.dll 5.00.2168.1 8.77 KB (8,976 bytes) 10/31/2002 2:21:13 AM
Microsoft Corporation c:\winnt\system32\igmpagnt.dll
acsmib.dll 5.00.2167.1 11.27 KB (11,536 bytes) 10/31/2002 2:17:25 AM
Microsoft Corporation c:\winnt\system32\acsmib.dll
evntagnt.dll 5.00.2195.4874 95.27 KB (97,552 bytes)
6/25/2002 4:07:04 AM Microsoft Corporation
c:\winnt\system32\evntagnt.dll
snmpmib.dll 5.00.2134.1 5.77 KB (5,904 bytes) 10/31/2002 2:44:21 AM
Microsoft Corporation c:\winnt\system32\snmpmib.dll
hostmib.dll 5.00.2195.4874 36.27 KB (37,136 bytes)
6/25/2002 4:07:05 AM Microsoft Corporation
c:\winnt\system32\hostmib.dll
inetmib1.dll 5.00.2195.4874 28.77 KB (29,456 bytes)
6/25/2002 4:07:06 AM Microsoft Corporation
c:\winnt\system32\inetmib1.dll

lmmib2.dll 5.00.2195.4874 29.27 KB (29,968 bytes)
6/25/2002 4:07:08 AM Microsoft Corporation
snmpapi.dll 5.00.2195.4874 17.27 KB (17,680 bytes)
6/25/2002 4:07:20 AM Microsoft Corporation
c:\winnt\system32\snmpapi.dll
snmp.exe 5.00.2195.5080 29.77 KB (30,480 bytes)
10/31/2002 2:44:21 AM Microsoft Corporation
c:\winnt\system32\snmp.exe
msidle.dll 5.00.2920.0000 6.27 KB (6,416 bytes)
10/31/2002 2:23:06 AM Microsoft Corporation
c:\winnt\system32\msidle.dll
mstask.exe 4.71.2195.1 115.77 KB (118,544 bytes) 6/25/2002 4:07:12 AM
Microsoft Corporation c:\winnt\system32\mstask.exe
regsvc.exe 5.00.2195.3649 65.27 KB (66,832 bytes)
6/25/2002 4:07:17 AM Microsoft Corporation
c:\winnt\system32\regsvc.exe
llsrpc.dll 5.00.2195.4907 47.77 KB (48,912 bytes)
10/31/2002 2:22:07 AM Microsoft Corporation
c:\winnt\system32\llsrpc.dll
llssrv.exe 5.00.2195.4907 81.27 KB (83,216 bytes)
7/22/2002 12:05:04 PM Microsoft Corporation
c:\winnt\system32\llssrv.exe
wmi.dll 5.00.2191.1 6.27 KB (6,416 bytes) 10/31/2002 2:27:51 AM
Microsoft Corporation c:\winnt\system32\wmi.dll
netshell.dll 5.00.2195.5431 457.77 KB (468,752 bytes)
6/25/2002 4:07:14 AM Microsoft Corporation
c:\winnt\system32\netshell.dll
netman.dll 5.00.2195.5282 89.27 KB (91,408 bytes)
6/25/2002 4:07:14 AM Microsoft Corporation
c:\winnt\system32\netman.dll
comsvcs.dll 2000.2.3497.0 1.37 MB (1,439,504 bytes)
6/25/2002 4:07:00 AM Microsoft Corporation
c:\winnt\system32\comsvcs.dll
ntmsdba.dll 5.00.2195.5279 169.27 KB (173,328 bytes)
6/25/2002 4:07:15 AM Microsoft Corporation
c:\winnt\system32\ntmsdba.dll
rasdlg.dll 5.00.2195.5438 515.77 KB (528,144 bytes)
10/31/2002 2:24:53 AM Microsoft Corporation
c:\winnt\system32\rasdlg.dll
netcfgx.dll 5.00.2195.4874 534.77 KB (547,600 bytes)
6/25/2002 4:07:13 AM Microsoft Corporation
c:\winnt\system32\netcfgx.dll
rasmans.dll 5.00.2195.5436 149.27 KB (152,848 bytes)
6/25/2002 4:07:17 AM Microsoft Corporation
c:\winnt\system32\rasmans.dll
sens.dll 5.00.2163.1 36.77 KB (37,648 bytes) 10/31/2002 2:25:25 AM
Microsoft Corporation c:\winnt\system32\sens.dll
ntmssvc.dll 5.00.2195.5254 391.77 KB (401,168 bytes)
6/25/2002 4:07:15 AM Microsoft Corporation
c:\winnt\system32\ntmssvc.dll
es.dll 2000.2.3497.0 225.27 KB (230,672 bytes)
6/25/2002 4:07:04 AM Microsoft Corporation
c:\winnt\system32\es.dll

mtxoci.dll 2000.2.3497.0 103.77 KB (106,256 bytes)
6/25/2002 4:07:13 AM Microsoft Corporation
c:\winnt\system32\mtxoci.dll
resutils.dll 5.00.2195.5339 39.77 KB (40,720 bytes)
6/25/2002 4:07:18 AM Microsoft Corporation
c:\winnt\system32\resutils.dll
clusapi.dll 5.00.2195.4678 54.27 KB (55,568 bytes)
6/25/2002 4:07:00 AM Microsoft Corporation
c:\winnt\system32\clusapi.dll
msvcp50.dll 5.00.7051 552.50 KB (565,760 bytes) 10/31/2002 2:23:24 AM Microsoft Corporation c:\winnt\system32\msvcp50.dll
xolehlp.dll 1999.9.3421.3 17.27 KB (17,680 bytes)
10/31/2002 2:44:23 AM Microsoft Corporation
c:\winnt\system32\xolehlp.dll
msdtclog.dll 2000.2.3497.0 86.77 KB (88,848 bytes)
6/25/2002 4:07:09 AM Microsoft Corporation
c:\winnt\system32\msdtclog.dll
mtxclu.dll 2000.2.3497.0 51.27 KB (52,496 bytes)
6/25/2002 4:07:13 AM Microsoft Corporation
c:\winnt\system32\mtxclu.dll
msdtcprx.dll 2000.2.3497.0 683.77 KB (700,176 bytes)
6/25/2002 4:07:09 AM Microsoft Corporation
c:\winnt\system32\msdtcprx.dll
txfaux.dll 2000.2.3497.0 383.27 KB (392,464 bytes)
6/25/2002 4:07:21 AM Microsoft Corporation
c:\winnt\system32\txfaux.dll
msdtctm.dll 2000.2.3497.0 1.08 MB (1,128,208 bytes)
6/25/2002 4:07:09 AM Microsoft Corporation
c:\winnt\system32\msdtctm.dll
msdtc.exe 1999.9.3421.3 6.77 KB (6,928 bytes)
10/31/2002 2:44:23 AM Microsoft Corporation
c:\winnt\system32\msdtc.exe
inetpp.dll 5.00.2195.4299 64.27 KB (65,808 bytes)
6/25/2002 4:07:06 AM Microsoft Corporation
c:\winnt\system32\inetpp.dll
win32spl.dll 5.00.2195.5201 92.27 KB (94,480 bytes)
10/31/2002 2:27:38 AM Microsoft Corporation
c:\winnt\system32\win32spl.dll
usbmon.dll 5.00.2195.4299 11.27 KB (11,536 bytes)
6/25/2002 4:07:22 AM Microsoft Corporation
c:\winnt\system32\usbmon.dll
tcpmon.dll 5.00.2195.4299 40.77 KB (41,744 bytes)
6/25/2002 4:07:21 AM Microsoft Corporation
c:\winnt\system32\tcpmon.dll
pjlmon.dll 5.00.2165.1 12.77 KB (13,072 bytes) 11/30/1999 3:39:36 PM Microsoft Corporation c:\winnt\system32\pjlmon.dll
cnbjmon.dll 5.00.2134.1 43.77 KB (44,816 bytes) 11/30/1999 3:38:48 PM Microsoft Corporation c:\winnt\system32\cnbjmon.dll
localspl.dll 5.00.2195.5423 250.27 KB (256,272 bytes)
10/31/2002 2:22:09 AM Microsoft Corporation
c:\winnt\system32\localspl.dll

spoolss.dll 5.00.2195.5400 61.77 KB (63,248 bytes)
10/31/2002 2:38:21 AM Microsoft Corporation
spoolsv.exe 5.00.2195.4299 44.27 KB (45,328 bytes)
10/31/2002 2:38:21 AM Microsoft Corporation
c:\winnt\system32\spoolsv.exe
clbcatq.dll 2000.2.3497.0 497.77 KB (509,712 bytes)
6/25/2002 4:07:00 AM Microsoft Corporation
c:\winnt\system32\clbcatq.dll
rpcss.dll 5.00.2195.5429 231.27 KB (236,816 bytes)
6/25/2002 4:07:18 AM Microsoft Corporation
c:\winnt\system32\rpcss.dll
svchost.exe 5.00.2134.1 7.77 KB (7,952 bytes) 10/31/2002 2:26:11 AM Microsoft Corporation c:\winnt\system32\svchost.exe
rdpwsx.dll 5.00.2195.5243 97.90 KB (100,248 bytes)
6/25/2002 4:07:17 AM Microsoft Corporation
c:\winnt\system32\rdpwsx.dll
mstlsapi.dll 5.00.2195.3895 25.77 KB (26,384 bytes)
6/25/2002 4:07:13 AM Microsoft Corporation
c:\winnt\system32\mstlsapi.dll
icaapi.dll 5.00.2195.3895 122.77 KB (125,712 bytes)
6/25/2002 4:07:05 AM Microsoft Corporation
c:\winnt\system32\icaapi.dll
regapi.dll 5.00.2195.5201 35.27 KB (36,112 bytes)
6/25/2002 4:07:17 AM Microsoft Corporation
c:\winnt\system32\regapi.dll
termsrv.exe 5.00.2195.5276 138.77 KB (142,096 bytes)
6/25/2002 4:07:21 AM Microsoft Corporation
c:\winnt\system32\termsrv.exe
iissuba.dll 5.00.0984 9.77 KB (10,000 bytes) 10/31/2002 2:21:22 AM Microsoft Corporation c:\winnt\system32\iissuba.dll
dssenh.dll 5.00.2195.3665 142.77 KB (146,192 bytes)
6/25/2002 4:07:33 AM Microsoft Corporation
c:\winnt\system32\dssenh.dll
oakley.dll 5.00.2195.5326 382.27 KB (391,440 bytes)
6/25/2002 4:07:15 AM Microsoft Corporation
c:\winnt\system32\oakley.dll
mfc42u.dll 6.00.8665.0 972.05 KB (995,384 bytes) 10/31/2002 2:22:34 AM Microsoft Corporation c:\winnt\system32\mfc42u.dll
polagent.dll 5.00.2195.5428 94.77 KB (97,040 bytes)
6/25/2002 4:07:16 AM Microsoft Corporation
c:\winnt\system32\polagent.dll
scecli.dll 5.00.2195.4874 109.27 KB (111,888 bytes)
6/25/2002 4:07:18 AM Microsoft Corporation
c:\winnt\system32\scecli.dll
atl.dll 3.00.9435 73.06 KB (74,810 bytes) 6/25/2002 4:06:58 AM Microsoft Corporation c:\winnt\system32\atl.dll
certcli.dll 5.00.2195.3649 130.27 KB (133,392 bytes)
6/25/2002 4:06:59 AM Microsoft Corporation
c:\winnt\system32\certcli.dll
esent.dll 6.0.3940.25 1.09 MB (1,137,936 bytes) 6/25/2002 4:07:04 AM Microsoft Corporation c:\winnt\system32\esent.dll

ntdsatq.dll 5.00.2195.5246 31.27 KB (32,016 bytes)
6/25/2002 4:07:14 AM Microsoft Corporation
ntdsa.dll 5.00.2195.5438 1002.27 KB (1,026,320 bytes)
6/25/2002 4:07:14 AM Microsoft Corporation
c:\winnt\system32\ntdsa.dll
kdcsvc.dll 5.00.2195.5246 141.77 KB (145,168 bytes)
6/25/2002 4:07:08 AM Microsoft Corporation
c:\winnt\system32\kdcsvc.dll
sfmapi.dll 5.00.2134.1 38.77 KB (39,696 bytes) 10/31/2002 2:25:32 AM Microsoft Corporation c:\winnt\system32\sfmapi.dll
rassfm.dll 5.00.2195.4874 21.27 KB (21,776 bytes)
6/25/2002 4:07:17 AM Microsoft Corporation
c:\winnt\system32\rassfm.dll
rsabase.dll 5.00.2195.3839 128.27 KB (131,344 bytes)
7/22/2002 12:05:04 PM Microsoft Corporation
c:\winnt\system32\rsabase.dll
schannel.dll 5.00.2195.5284 139.27 KB (142,608 bytes)
10/31/2002 2:25:19 AM Microsoft Corporation
c:\winnt\system32\schannel.dll
netlogon.dll 5.00.2195.5400 362.77 KB (371,472 bytes)
6/25/2002 4:07:14 AM Microsoft Corporation
c:\winnt\system32\netlogon.dll
kerberos.dll 5.00.2195.5246 202.77 KB (207,632 bytes)
6/25/2002 4:07:08 AM Microsoft Corporation
c:\winnt\system32\kerberos.dll
msprivs.dll 5.00.2154.1 41.50 KB (42,496 bytes) 10/31/2002 2:23:19 AM Microsoft Corporation c:\winnt\system32\msprivs.dll
samsrv.dll 5.00.2195.5201 374.27 KB (383,248 bytes)
10/31/2002 2:25:15 AM Microsoft Corporation
c:\winnt\system32\samsrv.dll
lsasrv.dll 5.00.2195.5430 500.27 KB (512,272 bytes)
10/31/2002 2:22:13 AM Microsoft Corporation
c:\winnt\system32\lsasrv.dll
lsass.exe 5.00.2195.5430 32.77 KB (33,552 bytes)
10/31/2002 2:22:13 AM Microsoft Corporation
c:\winnt\system32\lsass.exe
xactsrv.dll 5.00.2134.1 90.27 KB (92,432 bytes) 10/31/2002 2:28:07 AM Microsoft Corporation c:\winnt\system32\xactsrv.dll
ntlsapi.dll 5.00.2195.4907 6.77 KB (6,928 bytes)
10/31/2002 2:24:05 AM Microsoft Corporation
c:\winnt\system32\ntlsapi.dll
wmicore.dll 5.00.2195.3649 72.27 KB (74,000 bytes)
6/25/2002 4:07:23 AM Microsoft Corporation
c:\winnt\system32\wmicore.dll
rasadhlp.dll 5.00.2168.1 7.27 KB (7,440 bytes) 10/31/2002 2:24:50 AM Microsoft Corporation c:\winnt\system32\rasadhlp.dll
winrnr.dll 5.00.2160.1 18.77 KB (19,216 bytes) 10/31/2002 2:27:47 AM Microsoft Corporation c:\winnt\system32\winrnr.dll
rnr20.dll 5.00.2195.4874 35.77 KB (36,624 bytes)
6/25/2002 4:07:18 AM Microsoft Corporation
c:\winnt\system32\rnr20.dll

mswsock.dll 5.00.2195.4874 62.77 KB (64,272 bytes)
6/25/2002 4:07:13 AM Microsoft Corporation
c:\winnt\system32\mswsock.dll
msgsvc.dll 5.00.2195.4874 34.77 KB (35,600 bytes)
10/31/2002 2:23:03 AM Microsoft Corporation
c:\winnt\system32\msgsvc.dll
browser.dll 5.00.2195.4874 48.77 KB (49,936 bytes)
6/25/2002 4:06:58 AM Microsoft Corporation
c:\winnt\system32\browser.dll
alrsvc.dll 5.00.2134.1 17.77 KB (18,192 bytes) 10/31/2002 2:17:56 AM
Microsoft Corporation c:\winnt\system32\alrsvc.dll
trkwks.dll 5.00.2195.4874 88.77 KB (90,896 bytes)
6/25/2002 4:07:21 AM Microsoft Corporation
c:\winnt\system32\trkwks.dll
seclogon.dll 5.00.2195.5201 17.27 KB (17,680 bytes)
6/25/2002 4:07:18 AM Microsoft Corporation
c:\winnt\system32\seclogon.dll
psbase.dll 5.00.2195.4822 111.77 KB (114,448 bytes)
6/25/2002 4:07:17 AM Microsoft Corporation
c:\winnt\system32\psbase.dll
cryptsvc.dll 5.00.2195.4368 73.27 KB (75,024 bytes)
6/25/2002 4:07:01 AM Microsoft Corporation
c:\winnt\system32\cryptsvc.dll
cryptdll.dll 5.00.2135.1 41.27 KB (42,256 bytes) 10/31/2002 2:18:43 AM
Microsoft Corporation c:\winnt\system32\cryptdll.dll
wkssvc.dll 5.00.2195.4874 95.27 KB (97,552 bytes)
10/31/2002 2:27:51 AM Microsoft Corporation
c:\winnt\system32\wkssvc.dll
srvsvc.dll 5.00.2195.5400 81.77 KB (83,728 bytes)
10/31/2002 2:26:03 AM Microsoft Corporation
c:\winnt\system32\srvsvc.dll
cfgmgr32.dll 5.00.2134.1 16.77 KB (17,168 bytes) 10/31/2002 2:18:23 AM
Microsoft Corporation c:\winnt\system32\cfgmgr32.dll
dmserver.dll 2195.3649.297.3 12.27 KB (12,560 bytes)
6/25/2002 4:07:03 AM VERITAS Software Corp.
c:\winnt\system32\dmserver.dll
wshtcpip.dll 5.00.2195.4874 17.27 KB (17,680 bytes)
6/25/2002 4:07:23 AM Microsoft Corporation
c:\winnt\system32\wshtcpip.dll
msafd.dll 5.00.2195.4874 103.27 KB (105,744 bytes)
6/25/2002 4:07:09 AM Microsoft Corporation
c:\winnt\system32\msafd.dll
lmhsvc.dll 5.00.2195.4874 9.77 KB (10,000 bytes)
10/31/2002 2:22:07 AM Microsoft Corporation
c:\winnt\system32\lmhsvc.dll
dnsrslvr.dll 5.00.2195.5354 89.77 KB (91,920 bytes)
6/25/2002 4:07:03 AM Microsoft Corporation
c:\winnt\system32\dnsrslvr.dll
tapi32.dll 5.00.2182.1 123.27 KB (126,224 bytes) 10/31/2002 2:26:19 AM
Microsoft Corporation c:\winnt\system32\tapi32.dll

rasman.dll 5.00.2195.5292 54.77 KB (56,080 bytes)
10/31/2002 2:24:53 AM Microsoft Corporation
rasapi32.dll 5.00.2195.5439 191.77 KB (196,368 bytes)
10/31/2002 2:24:51 AM Microsoft Corporation
c:\winnt\system32\rasapi32.dll
rtutils.dll 5.00.2168.1 43.77 KB (44,816 bytes) 10/31/2002 2:25:13 AM
Microsoft Corporation c:\winnt\system32\rtutils.dll
adsldpc.dll 5.00.2195.5400 127.77 KB (130,832 bytes)
6/25/2002 4:06:58 AM Microsoft Corporation
c:\winnt\system32\adsldpc.dll
activeds.dll 5.00.2195.5312 175.27 KB (179,472 bytes)
6/25/2002 4:06:55 AM Microsoft Corporation
c:\winnt\system32\activeds.dll
oleaut32.dll 2.40.4518 612.27 KB (626,960 bytes) 10/31/2002 2:24:22 AM
Microsoft Corporation c:\winnt\system32\oleaut32.dll
mprapi.dll 5.00.2181.1 79.27 KB (81,168 bytes) 10/31/2002 2:22:47 AM
Microsoft Corporation c:\winnt\system32\mprapi.dll
iphlpapi.dll 5.00.2195.2 68.27 KB (69,904 bytes) 6/25/2002 4:07:06 AM
Microsoft Corporation c:\winnt\system32\iphlpapi.dll
icmp.dll 5.00.2134.1 7.27 KB (7,440 bytes) 10/31/2002 2:21:11 AM
Microsoft Corporation c:\winnt\system32\icmp.dll
dhcpcsvc.dll 5.00.2195.4874 87.77 KB (89,872 bytes)
10/31/2002 2:18:53 AM Microsoft Corporation
c:\winnt\system32\dhcpcsvc.dll
eventlog.dll 5.00.2195.5336 44.27 KB (45,328 bytes)
6/25/2002 4:07:04 AM Microsoft Corporation
c:\winnt\system32\eventlog.dll
ntdsapi.dll 5.00.2195.4827 56.27 KB (57,616 bytes)
6/25/2002 4:07:14 AM Microsoft Corporation
c:\winnt\system32\ntdsapi.dll
scesrv.dll 5.00.2195.5316 242.77 KB (248,592 bytes)
6/25/2002 4:07:18 AM Microsoft Corporation
c:\winnt\system32\scesrv.dll
umpnpmgr.dll 5.00.2182.1 86.27 KB (88,336 bytes)
10/31/2002 2:26:54 AM Microsoft Corporation
c:\winnt\system32\umpnpmgr.dll
services.exe 5.00.2195.3940 86.77 KB (88,848 bytes)
10/31/2002 2:25:27 AM Microsoft Corporation
c:\winnt\system32\services.exe
msv1_0.dll 5.00.2195.4745 112.27 KB (114,960 bytes)
10/31/2002 2:23:23 AM Microsoft Corporation
c:\winnt\system32\msv1_0.dll
lz32.dll 5.00.2134.1 9.77 KB (10,000 bytes) 10/31/2002 2:22:14 AM
Microsoft Corporation c:\winnt\system32\lz32.dll
version.dll 5.00.2134.1 15.77 KB (16,144 bytes) 10/31/2002 2:27:08 AM
Microsoft Corporation c:\winnt\system32\version.dll
winspool.drv 5.00.2195.5225 111.27 KB (113,936 bytes)
10/31/2002 2:27:49 AM Microsoft Corporation
c:\winnt\system32\winspool.drv
winscard.dll 5.00.2134.1 77.27 KB (79,120 bytes) 10/31/2002 2:27:48 AM
Microsoft Corporation c:\winnt\system32\winscard.dll

wlnotify.dll 5.00.2195.5377 54.27 KB (55,568 bytes)
6/25/2002 4:07:23 AM Microsoft Corporation
cscdll.dll 5.00.2195.5434 98.77 KB (101,136 bytes)
6/25/2002 4:07:01 AM Microsoft Corporation
c:\winnt\system32\cscdll.dll
rsaenh.dll 5.00.2195.3839 130.77 KB (133,904 bytes)
6/25/2002 4:07:33 AM Microsoft Corporation
c:\winnt\system32\rsaenh.dll
mscat32.dll 5.131.2134.1 7.77 KB (7,952 bytes) 10/31/2002 2:22:57 AM
Microsoft Corporation c:\winnt\system32\mscat32.dll
ole32.dll 5.00.2195.5400 968.27 KB (991,504 bytes)
6/25/2002 4:07:16 AM Microsoft Corporation
c:\winnt\system32\ole32.dll
imagehlp.dll 5.00.2195.5242 125.77 KB (128,784 bytes)
10/31/2002 2:21:24 AM Microsoft Corporation
c:\winnt\system32\imagehlp.dll
msasn1.dll 5.00.2195.4067 51.27 KB (52,496 bytes)
10/31/2002 2:22:57 AM Microsoft Corporation
c:\winnt\system32\msasn1.dll
crypt32.dll 5.131.2195.4558 464.27 KB (475,408 bytes)
6/25/2002 4:07:01 AM Microsoft Corporation
c:\winnt\system32\crypt32.dll
wintrust.dll 5.131.2195.3775 162.27 KB (166,160 bytes)
6/25/2002 4:07:23 AM Microsoft Corporation
c:\winnt\system32\wintrust.dll
mpr.dll 5.00.2195.3649 53.77 KB (55,056 bytes)
6/25/2002 4:07:09 AM Microsoft Corporation
c:\winnt\system32\mpr.dll
shlwapi.dll 5.00.3502.5332 283.27 KB (290,064 bytes)
6/25/2002 4:07:19 AM Microsoft Corporation
c:\winnt\system32\shlwapi.dll
shell32.dll 5.00.3502.5436 2.26 MB (2,374,416 bytes)
6/25/2002 4:07:19 AM Microsoft Corporation
c:\winnt\system32\shell32.dll
msgina.dll 5.00.2195.4733 324.77 KB (332,560 bytes)
10/31/2002 2:23:03 AM Microsoft Corporation
c:\winnt\system32\msgina.dll
comctl32.dll 5.81 539.27 KB (552,208 bytes) 10/31/2002 2:18:34 AM
Microsoft Corporation c:\winnt\system32\comctl32.dll
setupapi.dll 5.00.2195.5400 553.77 KB (567,056 bytes)
10/31/2002 2:25:29 AM Microsoft Corporation
c:\winnt\system32\setupapi.dll
winmm.dll 5.00.2161.1 184.77 KB (189,200 bytes) 10/31/2002 2:27:46 AM
Microsoft Corporation c:\winnt\system32\winmm.dll
winsta.dll 5.00.2195.4655 36.77 KB (37,648 bytes)
6/25/2002 4:07:23 AM Microsoft Corporation
c:\winnt\system32\winsta.dll
wsock32.dll 5.00.2195.4874 21.27 KB (21,776 bytes)
6/25/2002 4:07:23 AM Microsoft Corporation
c:\winnt\system32\wsock32.dll
dnsapi.dll 5.00.2195.5354 131.27 KB (134,416 bytes)
6/25/2002 4:07:03 AM Microsoft Corporation
c:\winnt\system32\dnsapi.dll

wldap32.dll 5.00.2195.5400 158.77 KB (162,576 bytes)
6/25/2002 4:07:23 AM Microsoft Corporation
c:\winnt\system32\wldap32.dll
ws2help.dll 5.00.2134.1 17.77 KB (18,192 bytes) 10/31/2002 2:28:03 AM Microsoft Corporation c:\winnt\system32\ws2help.dll
ws2_32.dll 5.00.2195.4874 66.77 KB (68,368 bytes)
6/25/2002 4:07:23 AM Microsoft Corporation
c:\winnt\system32\ws2_32.dll
samlib.dll 5.00.2195.4827 49.77 KB (50,960 bytes)
10/31/2002 2:25:15 AM Microsoft Corporation
c:\winnt\system32\samlib.dll
netrap.dll 5.00.2134.1 11.27 KB (11,536 bytes) 10/31/2002 2:23:41 AM Microsoft Corporation c:\winnt\system32\netrap.dll
netapi32.dll 5.00.2195.5427 305.27 KB (312,592 bytes)
6/25/2002 4:07:13 AM Microsoft Corporation
c:\winnt\system32\netapi32.dll
profmap.dll 5.00.2181.1 29.27 KB (29,968 bytes) 10/31/2002 2:24:42 AM Microsoft Corporation c:\winnt\system32\profmap.dll
secur32.dll 5.00.2195.4587 47.27 KB (48,400 bytes)
6/25/2002 4:07:18 AM Microsoft Corporation
c:\winnt\system32\secur32.dll
sfc.dll 5.00.2195.3649 92.11 KB (94,320 bytes)
6/25/2002 4:07:19 AM Microsoft Corporation
c:\winnt\system32\sfc.dll
nddeapi.dll 5.00.2195.4509 15.77 KB (16,144 bytes)
6/25/2002 4:07:13 AM Microsoft Corporation
c:\winnt\system32\nddeapi.dll
userenv.dll 5.00.2195.5425 363.77 KB (372,496 bytes)
10/31/2002 2:27:00 AM Microsoft Corporation
c:\winnt\system32\userenv.dll
user32.dll 5.00.2195.4314 395.77 KB (405,264 bytes)
10/31/2002 2:27:01 AM Microsoft Corporation
c:\winnt\system32\user32.dll
gdi32.dll 5.00.2195.5252 228.77 KB (234,256 bytes)
10/31/2002 2:20:49 AM Microsoft Corporation
c:\winnt\system32\gdi32.dll
rpcrt4.dll 5.00.2195.5419 440.27 KB (450,832 bytes)
6/25/2002 4:07:18 AM Microsoft Corporation
c:\winnt\system32\rpcrt4.dll
advapi32.dll 5.00.2195.5385 358.77 KB (367,376 bytes)
10/31/2002 2:17:53 AM Microsoft Corporation
c:\winnt\system32\advapi32.dll
kernel32.dll 5.00.2195.5400 716.77 KB (733,968 bytes)
10/31/2002 2:22:00 AM Microsoft Corporation
c:\winnt\system32\kernel32.dll
msvcrt.dll 6.10.9359.0 284.05 KB (290,869 bytes) 7/22/2002 12:05:04 PM Microsoft Corporation c:\winnt\system32\msvcrt.dll
winlogon.exe 5.00.2195.5386 174.77 KB (178,960 bytes)
6/25/2002 4:07:22 AM Microsoft Corporation
c:\winnt\system32\winlogon.exe
sfcfiles.dll 5.00.2195.5426 951.27 KB (974,096 bytes)
6/25/2002 4:07:19 AM Microsoft Corporation
ntdll.dll 5.00.2195.5400 479.27 KB (490,768 bytes)
10/31/2002 2:23:56 AM Microsoft Corporation
c:\winnt\system32\ntdll.dll
smss.exe 5.00.2195.5382 44.77 KB (45,840 bytes)
10/31/2002 2:25:47 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			
Background Intelligent Transfer Service	BITS	Stopped	Manual	Share Process
c:\winnt\system32\svchost.exe -k bitsgroup			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			
Compaq NIC Agents	CPQNicMgmt	Stopped	Disabled	Own Process
c:\winnt\system32\cpqnimgt\cpqnimgt.exe			Normal	
LocalSystem	0			
Compaq Remote Monitor Service	CpqRcmc	Stopped	Disabled	Own Process
c:\winnt\system32\cpqrcmc.exe			Normal	
LocalSystem	0			
Compaq Version Control Agent	cpqvcagent	Stopped	Disabled	Own Process
c:\compaq\vcagent\vcagent.exe			Normal	
LocalSystem	0			
Compaq Web Agent	CpqWebMgmt	Stopped	Disabled	Own Process
c:\winnt\system32\cpqmgmt\cpqwmgmt.exe			Normal	
LocalSystem	0			
Compaq Foundation Agents	CqMgHost	Stopped	Disabled	Own Process
c:\winnt\system32\cpqmgmt\cqmghost\cqmghost.exe			Normal	
LocalSystem	0			
Compaq Server Agents	CqMgServ	Stopped	Disabled	Own Process
c:\winnt\system32\cpqmgmt\cqmgserv\cqmgserv.exe			Normal	
LocalSystem	0			
Compaq Storage Agents	CqMgStor	Stopped	Disabled	Own Process
c:\winnt\system32\cpqmgmt\cqmgstor\cqmgstor.exe			Normal	
LocalSystem	0			
Distributed File System	Dfs	Running	Auto	Own Process
c:\winnt\system32\dfssvc.exe			Normal	
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			
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 Command Service RMSYS Running Auto Own
Process c:\program files\benchcraft\rsys.exe Normal
        \Administrator 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 RSVP RSVP Running Manual 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
SNMP Service SNMP Running Auto Own
Process c:\winnt\system32\snmp.exe Normal
        LocalSystem 0
SNMP Trap Service SNMPTRAP Stopped Manual Own
Process c:\winnt\system32\snmptrap.exe Normal
        LocalSystem 0
Print Spooler Spooler Running Auto Own Process
        c:\winnt\system32\spoolsv.exe Normal
        LocalSystem 0
Surveyor Surveyor Running Auto Own Process
        c:\compaq\survey\surveyor.exe Normal
        LocalSystem 0
Compaq System Shutdown Service sysdown Stopped Disabled
        Own Process c:\winnt\system32\sysdown.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 Running Auto 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
Automatic Updates wuauserv Running Auto Share
Process c:\winnt\system32\svchost.exe -k wugroup 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
Compaq System Tools All Users:Compaq System Tools All Users
Microsoft SQL Server All Users:Microsoft SQL Server All Users
Startup All Users:Startup All Users
Accessories TPCWEB1\Administrator:Accessories
TPCWEB1\Administrator
Accessories\Accessibility TPCWEB1\Administrator:Accessories\Accessibility
TPCWEB1\Administrator
Accessories\Entertainment
TPCWEB1\Administrator:Accessories\Entertainment
TPCWEB1\Administrator
Accessories\System Tools TPCWEB1\Administrator:Accessories\System Tools
TPCWEB1\Administrator
Administrative Tools TPCWEB1\Administrator:Administrative Tools
TPCWEB1\Administrator
Benchcraft TPCWEB1\Administrator:Benchcraft
TPCWEB1\Administrator
Compaq Products and Services TPCWEB1\Administrator:Compaq Products and Services
TPCWEB1\Administrator
SANblade Control VIX TPCWEB1\Administrator:SANblade Control VIX
TPCWEB1\Administrator
Startup TPCWEB1\Administrator:Startup
TPCWEB1\Administrator

[Startup Programs]

Program Command User Name Location
No startup program information

[OLE Registration]

Object Local Server
Sound (OLE2) sndrec32.exe
Media Clip mplay32.exe
Video Clip mplay32.exe /avi
MIDI Sequence mplay32.exe /mid
Sound Not Available
Media Clip Not Available
Image Document "C:\Program Files\Windows NT\Accessories\ImageVue\KodakImg.exe"
WordPad Document "%ProgramFiles%\Windows NT\Accessories\WORDPAD.EXE"
Windows Media Services DRM Storage object Not Available
Bitmap Image mspaint.exe

[Internet Explorer 5]

[Following are sub-categories of this main category]
[Summary]

Item Value
Version 5.00.3502.1000
Build 53502.1000
Product ID 51876-270-0607125-05823
Application Path C:\Program Files\Internet Explorer
Language English (United States)
Active Printer Not Available

Cipher Strength 168-bit
Content Advisor Disabled
IEAK Install No

[File Versions]

File	Version	Size	Date	Path	Company
advapi32.dll	5.0.2195.5385	359 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
advpack.dll	5.0.3502.4373	87 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
browselc.dll	5.0.3502.4373	35 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
browseui.dll	5.0.3502.4373	791 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
ckcnv.exe	5.0.2189.1	9 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
comctl32.dll	5.81.3315.3727	539 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
crypt32.dll	5.131.2195.4558	464 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
enhsig.dll	<File Missing>	Not Available	Not Available	Not Available	
iemigrat.dll	<File Missing>	Not Available	Not Available	Not Available	
iesetup.dll	5.0.3502.4373	57 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
iexplore.exe	5.0.2920.0	59 KB	12/7/1999 4:00:00 AM	C:\Program Files\Internet Explorer	Microsoft Corporation
imagehlp.dll	5.0.2195.5242	126 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
imghelp.dll	<File Missing>	Not Available	Not Available	Not Available	
inseng.dll	5.0.3502.4373	72 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
jobexec.dll	5.0.0.1	47 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
jscript.dll	5.1.0.5907	476 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
jsproxy.dll	5.0.2920.0	13 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
msaahtml.dll	<File Missing>	Not Available	Not Available	Not Available	
mshtml.dll	5.0.3502.5390	2284 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
msjava.dll	5.0.3805.0	924 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
msoss.dll	<File Missing>	Not Available	Not Available	Not Available	
msxml.dll	8.0.6730.0	494 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
occache.dll	5.0.3315.3727	86 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
ole32.dll	5.0.2195.5400	968 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
oleaut32.dll	2.40.4518.0	612 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
olepro32.dll	5.0.4518.0	160 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
rsabase.dll	5.0.2195.3839	128 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
rsaenh.dll	5.0.2195.3839	131 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
rsapi32.dll	<File Missing>	Not Available	Not Available	Not Available	
rsasig.dll	<File Missing>	Not Available	Not Available	Not Available	
schannel.dll	5.1.2195.0	139 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
shdoc401.dll	<File Missing>	Not Available	Not Available	Not Available	
shdocvw.dll	5.0.3502.5400	1079 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
shell32.dll	5.0.3502.5436	2319 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
shlwapi.dll	5.0.3502.5332	283 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
url.dll	5.0.3502.4510	82 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
urlmon.dll	5.0.3502.5400	442 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
vbscript.dll	5.1.0.7426	428 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
webcheck.dll	5.0.3315.3727	251 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
win.com	5.0.2134.1	24 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
wininet.dll	5.0.3502.4619	451 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
winsock.dll	3.10.0.103	3 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
wintrust.dll	5.131.2195.3775	162 KB	7/22/2002 11:05:04 AM	C:\WINNT\system32	Microsoft Corporation
wsock.vxd	<File Missing>	Not Available	Not Available	Not Available	

wsock32.dll 5.0.2195.4874 21 KB 7/22/2002 11:05:04 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 Disabled
AutoConfigURL
Proxy Disabled
ProxyServer
ProxyOverride

[Cache]

[Following are sub-categories of this main category]

[Summary]

Item Value
Page Refresh Type Automatic
Temporary Internet Files Folder C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files
Total Disk Space 17316 MB
Available Disk Space 14366 MB
Maximum Cache Size 541 MB
Available Cache Size 541 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 6/25/2002 to 6/1/2102 sha1RSA

[Other People Certificates]

Issued To Issued By Validity Signature Algorithm
No other people certificate information available

[Publishers]

Name
No publisher information available

[Security]

Zone Security Level
Local intranet Medium-low
Trusted sites Low
Internet Medium
Restricted sites High

C.3 RTE Input Parameters

3Tier.pro

Profile: 3Tier
File Path: C:\Benchcraft\3Tier.pro
Version: 3

Number of Engines: 20

Name: DRIVER1
Description:
Directory: c:\1.log
Machine: tpccl1.hp-perf.net
Parameter Set: Opt
Index: 0
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER16481093
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER2
Description:
Directory: c:\2.log
Machine: tpccl1.hp-perf.net
Parameter Set: Opt
Index: 100000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER26496421
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER3
Description:
Directory: c:\3.log
Machine: tpccl2.hp-perf.net
Parameter Set: Opt
Index: 200000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER317477109
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER4
Description:
Directory: c:\4.log
Machine: tpccl2.hp-perf.net
Parameter Set: Opt
Index: 300000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER417501687
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER5
Description:
Directory: c:\5.log
Machine: tpccl3.hp-perf.net
Parameter Set: Opt
Index: 400000000
Seed: 71524
Configured Users: 3500

Pipe Name: DRIVER513270750
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER6
Description:
Directory: c:\6.log
Machine: tpccl3.hp-perf.net
Parameter Set: Opt
Index: 500000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER613284796
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER7
Description:
Directory: c:\7.log
Machine: tpccl4.hp-perf.net
Parameter Set: Opt
Index: 600000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER713300515
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER8
Description:
Directory: c:\8.log
Machine: tpccl4.hp-perf.net
Parameter Set: Opt
Index: 700000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER813311312
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER9
Description:
Directory: c:\9.log
Machine: tpccl5.hp-perf.net
Parameter Set: Opt
Index: 800000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER9359269953
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER10
Description:
Directory: c:\10.log
Machine: tpccl5.hp-perf.net
Parameter Set: Opt
Index: 900000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER10359292890
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER11
Description:
Directory: c:\11.log
Machine: tpccl6.hp-perf.net
Parameter Set: Opt
Index: 1000000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER11426093468
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER12
Description:
Directory: c:\12.log
Machine: tpccl6.hp-perf.net
Parameter Set: Opt
Index: 1100000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER12426111203
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER13
Description:
Directory: c:\13.log
Machine: tpccl7.hp-perf.net
Parameter Set: Opt
Index: 1200000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER13426140656
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER14
Description:
Directory: c:\14.log
Machine: tpccl7.hp-perf.net
Parameter Set: Opt
Index: 1300000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER14426164718
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER15
Description:
Directory: c:\15.log
Machine: tpccl8.hp-perf.net
Parameter Set: Opt
Index: 1400000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER15426190531
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER16
Description:
Directory: c:\16.log
Machine: tpccl8.hp-perf.net
Parameter Set: Opt
Index: 1500000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER16426206921

Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER17
Description:
Directory: c:\17.log
Machine: tpccl9.hp-perf.net
Parameter Set: Opt
Index: 1600000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER17426242828
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER18
Description:
Directory: c:\18.log
Machine: tpccl9.hp-perf.net
Parameter Set: Opt
Index: 1700000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER18426254718
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER19
Description:
Directory: c:\19.log
Machine: tpccl10.hp-perf.net
Parameter Set: Opt
Index: 1800000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER19426276250
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER20
Description:
Directory: c:\20.log
Machine: tpccl10.hp-perf.net
Parameter Set: Opt
Index: 1900000000
Seed: 71524
Configured Users: 3500
Pipe Name: DRIVER20426289234
Connect Rate: 10
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Number of User groups: 20

Driver Engine: DRIVER1
IIS Server: tpcweb1.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1 - 350
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER2
IIS Server: tpcweb1.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 351 - 700
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER3
IIS Server: tpcweb1.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 701 - 1050
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER4
IIS Server: tpcweb1.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1051 - 1400
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER5
IIS Server: tpcweb2.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1401 - 1750
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER6
IIS Server: tpcweb2.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1751 - 2100
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER7
IIS Server: tpcweb2.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2101 - 2450
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER8
IIS Server: tpcweb2.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML

w_id Range: 2451 - 2800
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER9
IIS Server: tpcweb3.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2801 - 3150
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER10
IIS Server: tpcweb3.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3151 - 3500
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER11
IIS Server: tpcweb3.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3501 - 3850
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER12
IIS Server: tpcweb3.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3851 - 4200
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER13
IIS Server: tpcweb4.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4201 - 4550
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER14
IIS Server: tpcweb4.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4551 - 4900
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER15
IIS Server: tpcweb4.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4901 - 5250
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER16
IIS Server: tpcweb4.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5251 - 5600
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER17
IIS Server: tpcweb5.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5601 - 5950
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER18
IIS Server: tpcweb5.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5951 - 6300
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER19
IIS Server: tpcweb5.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 6301 - 6650
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Driver Engine: DRIVER20
IIS Server: tpcweb5.hp-perf.net
SQL Server: tpc-rx5670
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 6651 - 7000
w_id Min Warehouse: 1
w_id Max Warehouse: 7000
Scale: Normal
User Count: 3500
District id: 1
Scale Down: No

Number of Parameter Sets: 2

~Default
Default Parameter Set

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence
Menu					
Delay					
New Order	10.00	12.05	18.01		
0.10	5.00	0.10			
Payment	10.00	12.05	3.01		
0.10	5.00	0.10			
Delivery	1.00	5.05	2.01		
0.10	5.00	0.10			
Stock Level	1.00	5.05	2.01		
0.10	20.00	0.10			
Order Status	1.00	10.05	2.01		
0.10	5.00	0.10			

Opt

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence
Menu					
Delay					
New Order	44.95	12.05	18.01		
0.10	5.00	0.10			
Payment	43.01	12.05	3.01		
0.10	5.00	0.10			
Delivery	4.02	5.05	2.01		
0.10	5.00	0.10			
Stock Level	4.01	5.05	2.01		
0.10	20.00	0.10			
Order Status	4.02	10.05	2.01		
0.10	5.00	0.10			

Appendix D 60 Day Space Requirements

Note : Numbers are in KBytes unless otherwise specified						
Warehouses	7000	**tpmC**	87741.45	**tpmC/W**	12.53	
Table	**Rows**	**Data**	**Index**	**5% Space**	**8H Space**	**Total Space**
Warehouse	7,000	752	48	40		840
District	70,000	7,784	64	392		8,240
Item	100,000	9,528	64	221		9,813
New-order	63,000,000	966,048	2,312		560,000	1,528,360
History	210,000,000	11,666,680	40		1,961,607	13,628,327
Orders	210,000,000	6,436,784	2,927,000		1,574,398	10,938,182
Customer	210,000,000	152,727,280	9,106,752	3,722,183		165,556,215
Order-line	2,099,998,534	131,249,912	277,824		22,114,677	153,642,413
Stock	700,000,000	224,000,008	418,108	5,161,617		229,579,733
Totals		527,064,776	12,732,212	8,884,452	26,210,682	574,892,123
DB File Group	**Count**	**Size**	**Needed**	**Overhead**		**Not Needed**
MSSQL_misc_fg	5	212,480,000	181,553,737	1,815,537		29,110,725
MSSQL_cs_fg	5	415,744,000	399,087,307	3,990,873		12,665,820
Totals		628,224,000	580,641,044	5,806,410		41,776,545
Dynamic space	145,320,835	Sum of Data for Order, Order-Line and History (excluding free extents)				
Static space	409,167,016	Data + Index + 5% Space + Overhead - Dynamic space				
Free space	31,959,604	Total Seg. Size - Dynamic Space - Static Space - Not Needed				
Daily growth	29,144,367	(Dynamic space/W * 62.5)* tpmC				
Daily spread	(11,756,947)	Free space - 1.5 * Daily growth (zero if negative)				
60 day (KB)	2,157,829,064	Static space + 60 (daily growth + daily spread)				
60 day (GB)	2057.87	Excludes OS, Paging and RDBMS Logs				
Log size (MB)	89999.99	Total size of log file				
% Log used	79.82	% of log file used during entire run				
Total N-O Txn	15422723	Total count of N-O transactions during entire run				
Log per N-O txn	4.77	Number of Kbytes per New-Order transaction				
8 Hour Log (GB)	191.58	need double for mirroring				
os, file sys, swap	33.29					
	Disk size (GB)	**Priced Qty**	**Priced (GB)**	**Needed(GB)**	**Extra (GB)**	
Database, Sys	16.629	280	4656.12	2,091.15	2,598.25	
	33.287	1	33.29			
Mirrored Log	33.287	14	466.02	383.15	82.87	

Appendix E 3rd Party Pricing

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
http://www.microsoft.com/

Microsoft

December 2, 2002

Hewlett-Packard
Company
Larry Kemp
One Microsoft Way
Redmond, WA 98052

Mr. Kemp:

Here is the information you requested regarding pricing for several Microsoft products to be used in conjunction with your TPC-C benchmark testing.

All pricing shown is in US Dollars ($).

Part Number	Description	Unit Price	Quantity	Price
810-00560	**SQL Server 2000 Enterprise Edition (64-bit)** *Per processor licensing* *Discount Schedule: Open Program Level C* *Unit Price reflects a 17% discount from the retail unit price of $19,999.*	$16,541	4	$66,164
C11-00821	**Windows 2000 Server** *Server license only - No CALs* *Discount Schedule: Open Program - No Level* *Unit Price reflects a 8% discount from the retail unit price of $799.*	$738	5	$3,690
N/A	**Windows .Net Enterprise Server 2003** *Server license only - No CALs* *Discount Schedule: Open Program - No Level* *Unit Price reflects a 18% discount from the retail unit price of $6,599.*	$5,399	1	$5,399
254-00170	**Visual C++ .Net Standard** *No discounts applied*	$109	1	$109
PRO-PRORS-16U-01	**Database Server Support Package** *1 Year Term*	$1,950	3	$5,850

Some products may not be currently orderable but will be available through Microsoft's normal distribution channels by December 31, 2002.

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.

QLogic Corporation • 26600 Laguna Hills Drive • Aliso Viejo • CA 92656 • Ph: (949) 389-6000

To:	Gunter Zink Hewlett Packard One Microsoft Way Redmond, WA 98052	**Fax:**	
From:	Joann Laforge	**Date**:	December 12, 2002
Re:	MSRP for Qlogic HBA and Switch QLA2350 (single port HBA) Sanbox2-8 port switch	**Pages incl cover:**	1
cc:			

Gunter:

Qlogic is pleased to provide you the following MSRP for your TCP Benchmark publication.

Product	**Distributor**	**Price**
QLA2350	**Unique**	**$2,095**
QLA2352	**Unique**	**$3,595**
Sanbox2-8 port switch	**Unique**	**$7,995**
	Bell, Tech Data, Arrow	
Sanbox2/16 port switch	**Unique**	**$17,995**
	Bell, Tech Data, Arrow	

If you have any questions or need anything else, please let me know.

Thank you for your interest in Qlogic.

Joann Laforge

OEM Account Executive

Qlogic Corporation (281) 378-1565

Office: 281-378-1565

Cell: 281-513-9281

Fax: 281-378-1567

joann.laforge@qlogic.com

distribution or copying of this communication is strictly prohibited. If you have received this communication in error, please notify us immediately by telephone and return the original message to us at the above address via the US Postal Service. Thank You.

□ Acc Fax (949) 389-6187	□ Cust. Svce Fax (949) 389-6008	□ PPG Fax (949) 389-6210
□ CSG-1 Fax (949) 389-6308	□ HR Fax (949) 389-6110	□ Sales Fax (949) 389-6114
□ CSG-2 Fax (949) 389-6121	□ IT/Operations Fax (949) 389-6009	□ Shipping/Rcvg Fax (949) 389-6111
□ Corp. Mktg Fax (949) 389-6126	□ Mktg Fax (949) 389-6114	□ Tax Fax (949) 389-6487

Larry Kemp
Hewlett Packard
14335 NE 24th Street
Suite B-201

Hewlett Packard Company	**hp server rx5670**	HP Customer Business Center 8000 Foothill Boulevard Roseville, CA 95747 1-800-386-1117

Description	Price Key	Part Numbr	Unit Price	Qty	Extended Price
1GHz Itanium 2 w/ 3MB iL3 cache, 0 MB RAM, 0 disk	1	A6838A	$26,494	1	$26,494
CPU upgrade Itanium 2, 1GHz w/3MB iL3 cache	1	A6836A	$8,250	3	$24,750
4GB PC2100 DDR-SDRAM (4x1GB DIMMs)	1	A6834A	$8,000	12	$96,000
Memory Carrier Board	1	A6747A	$1,981	2	$3,962
HP 36GB, 15krpm Ultra320 hot-swap disk	1	A7049A	$996	2	$1,992
HP Rackmount Kit Factory	1	A5580A	$134	1	$134
DVD Rom drive	1	A5557B	$450	1	$450
Graphics USB Card	1	A6869A	$349	1	$349
HP USB keyboard and mouse	1	A7861A	$32	1	$32
HP Smart Array Controller 5304	2	*283551-B21*	$2,247	6	$13,482
5m LC to LC Cable Kit	2	221692-B22	$82	1	$82
2GB SFP Adapter Kit	2	221470-B21	$369	1	$369
S5500 15 carbon / silver monitor	2	261602-001	$139	1	$139
HP Rack Model 9142 (42U - Opal) - Flat Pallet	2	120663-B21	$1,352	2	$2,704
HP Power Distribution Unit 120-240V	1	E7671A	$146	3	$438
UPS R1500 XR	2	204404-001	$886	1	$886
HP Hardware Support 3 yr, 24x7, 4 hr rx5670	1	H4405Y#6BO	$7,052	1	$7,052
HP Hardware Support 3 yr, 24x7, 4 hr addt'l CPU	1	H4405Y#6BP	$1,153	3	$3,459
20/40 GB DAT Drive, External	1	C5687B	$1,300	1	$1,300
Storageworks enclosure 4314R	2	190209-001	$2,955	20	$59,100
18GB, 15krpm Ultra3 Wide disk	2	188122-B22	$390	280	$109,200
18GB, 15krpm Ultra3 Wide disk (10% spares)	2	188122-B22	$390	28	$10,920
Storageworks enclosure 4354R	2	190211-001	$3,523	1	$3,523
36GB, 15krpm Ultra3 Wide disk	2	232916-B22	$605	14	$8,470
36GB, 15krpm Ultra3 Wide disk (2 spares)	2	232916-B22	$605	2	$1,210
Hardware Support 3 yr, 24x7, 4hr empty enclosure	2	171242-002	$157	21	$3,297
HP ProLiant DL360R01 P1.4GHz 512KB 128MB	2	233271-001	$2,229	5	$11,145
1.40GHz PIII Processor Option Kit (DL360 G2)	2	233273-B21	$734	5	$3,670
1G PC133-MHz option Kit	2	201694-B21	$425	5	$2,125
HP Mouse	2	261602-001	$5	5	$25
HP Enhanced Keyboard	2	265977-001	$12	5	$60
S5500 15 carbon / silver monitor	2	261602-001	$139	5	$695
18GB, 15krpm Ultra3 Wide disk	2	188122-B22	$390	5	$1,950
5M LC to LC Cable Kit	2	221692-B22	$82	5	$410
2GB Small Form Pluggable Adapter Kit	2	221470-B21	$369	5	$1,845
FM-EL724-36 3YR 24X7 4HR 300 SERIES SVR	2	162657-002	$1,450	5	$7,250
HP's Large Configuration Discount *					**-$75,420**
			Total:		**$333,549**

Price Key: 1-HP at 22% discount, 2-HP at 16% discount	* All discounts are based on US list prices and for similar quantities and configurations