TPC Benchmark® C
Full Disclosure Report

HP NetServer LXr 8500
using Microsoft SQLServer 2000 Enterprise Edition
on Microsoft Windows 2000 Advanced Server

Third Edition
August 23rd, 2001

Third Edition - August 23rd, 2001
First Printing.

Printed in U.S.A., August 23rd, 2001

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark® C test conducted on the HP NetServer LXr 8500 in a client/server configuration, using Microsoft SQLServer 2000 Enterprise Edition and Microsoft COM+ Transaction Monitor. The operating system used for the benchmark was Microsoft Windows 2000 Advanced Server.

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 NetServer LXr 8500 system. The Standard System Summary is given below.

Company Name	System Name	Database Software	Operating System
Hewlett-Packard Company	HP NetServer LXr 8500	Microsoft SQLServer 2000 Enterprise Edition	Microsoft Windows 2000 Advanced Server
Total System Cost	**TPC-C Throughput**	**Price/Performance**	**Availability Date**
$435,038.00	43046.55 tpmC	$10.11 per tpmC	March 1st, 2001

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 Tom Sawyer of Performance Metrics, Inc. to verify compliance with the relevant TPC specifications. The upgrade to the TPC-C Revision 5.0 specification was audited by Lorna Livingtree of Performance Metrics, Inc.

Additional copies of this Full Disclosure Report can be obtained from either the Transaction Processing Performance Council or Hewlett-Packard Company at the following addresses:

Transaction Processing Performance Council (TPC)
c/o Shanley Public Relations
777 North First Street, Suite 600
San Jose, CA 95112, USA
Phone: (408) 295-8894, (408) 295-9768 fax

or

Hewlett-Packard Company
Enterprise NetServer Division
10955 Tantau Avenue
Cupertino, CA 95014-0770 USA
Attn: Alexander Carlton, MS 45NUH

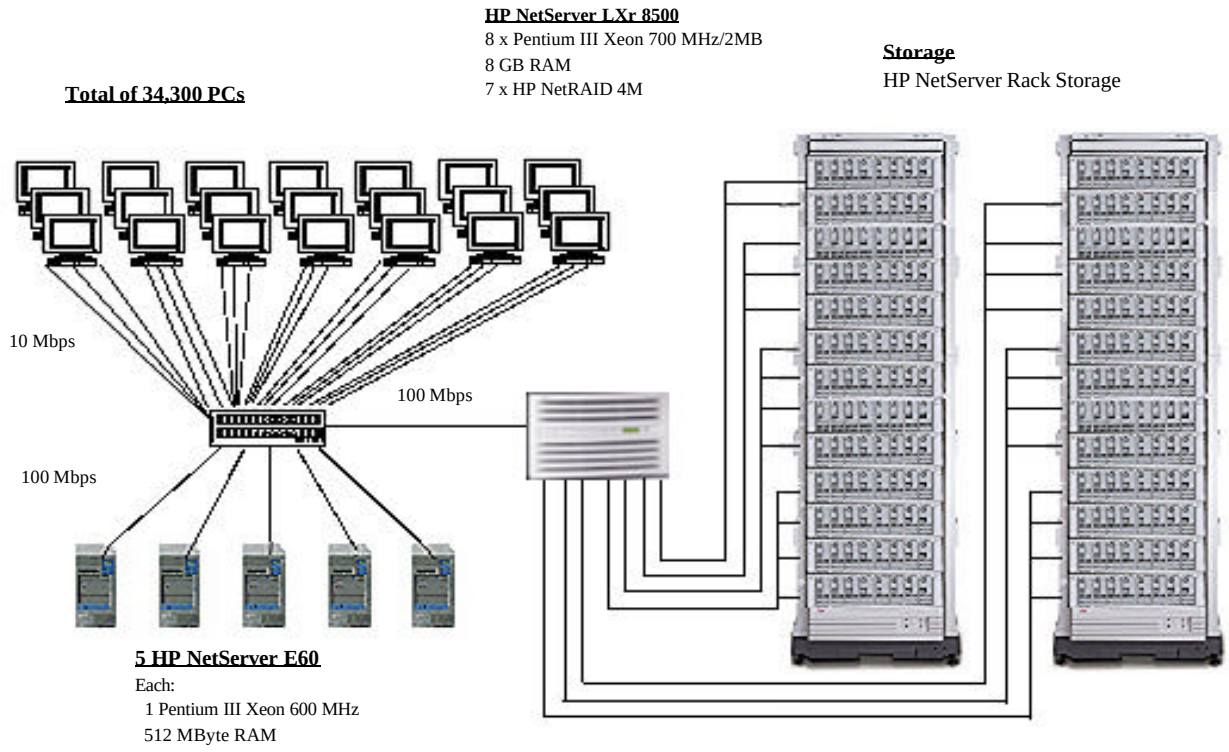
HP NetServer LXr 8500 Client/Server

TPC-C Rev 5.0

Report Date:
August 23rd, 2001

Total System Cost	TPC Throughput	Price/Performance	Availability Date
$435,038	43046.55 tpmC	$10.11 per tpmC	March 1st, 2001

Processors	Database Manager	Operating System	Other Software	Number of Users
8 Intel Pentium III Xeon 700MHz 2 Mbyte L2	Microsoft SQLServer 2000 Enterprise Edition	Microsoft Windows 2000 Advanced Server	Microsoft Visual C++ Microsoft COM+ Transaction Monitor	34300



System Components	Server		Each Client	
	Qty	Type	Qty	Type
Processors Cache Memory	8 each	700MHz Intel Pentium III Xeon 2 Mbyte L2 cache	1	600MHz Pentium III 512kbyte L2 Cache
Memory	16	512 Mbyte		512 Mbyte
Disk Controllers	7	HP NetRAID 4M	1	HP SCSI-2 Controller
Disk Drives	240	HP Hot-swap 9GB SCSI (8.47GB)	1	9 Gbyte disk
Total Storage	2920.06	Gbyte		8.46 Gbyte
Tape Drives	1	HP Surestore DAT24i		
Terminals	1	Console Terminal	1	Console Terminal

HP NetServer LXr 8500
8P/700-2MB/8GB

TPC-C Rev 5

Report Date: **23-Aug-01**

Description	Part Number	Brand	Price Key	Unit Price	Qty	Extended Price	Maint. Price
HP NetServer LXr 8500 Intel Pentium III Xeon 700MHz	P1765AV	HP	1	17,550	1	17,550	
HP LXr8500 Support Pack 3 years, 24x7, 4 hour response	H5515A	HP	2	2,193	1		2,193
Intel Pentium III Xeon 700MHz 2Mbyte L2	P1763AV	HP	1	3,829	7	26,803	
512MB Dimm for LXr 8500 (to 8GB)	D7138AV	HP	1	1,364	16	21,824	
HP NetRAID4M RAID controller w/64MB cache	D9161A	HP	1	1647	7	11,529	
HP 9GB, 10krpm Hot-swap disk module	P1168A	HP	1	262	1	262	
HP NetServer 10/100TX PCI LAN Adapter	D5013A	HP	1	69	1	69	
HP 15" VGA Monitor	D2828A	HP	1	185	1	185	
HP NetServer mini-DIN keyboard and mouse	D4950B/C375	HP	1	79	1	79	
HP Rack System/E41 (41 Units usable space)	J1500A	HP	1	1590	2	3,180	
HP Power Distribution Unit 120-240V	E5929A	HP	1	234	4	936	
HP SureStore DAT24i Internal Tape Drive	C1555D	HP	1	777	1	777	
APC Smart-UPS 3000 + Spares	588293	APC	1	1340	3	4,020	
					Server Hardware Subtotal	**87,214**	**2,193**
HP NetServer Rack Storage/12	D5989B	HP	1	1890	25	47,250	
HP RS/12 Support Pack 3 years, 24x7, 4 hour response	H5513A	HP	2	1450	25		36,250
HP 9GB, 10krpm Hot-swap disk module	P1168A	HP	1	262	240	62,880	
HP 18GB, 10krpm Hot-swap disk module	P1166A	HP	1	530	52	27,560	
HP SCSI Cable 2.5m UDHTS 68/HDTS 68	D6020A	HP	1	84	25	2,100	
					Storage Subtotal	**139,790**	**36,250**
Microsoft Windows 2000 Advanced Server, 25 Licences	C10-00475	MS	1	2399	1	2,399	
Microsoft SQL Server 2000 Enterprise Edition per processor license unlimited users (open program level C)	810-00846	MS	1	16541	8	132,328	6,285
					Server Software Subtotal	**134,727**	**6,285**
HP Netserver E60 Pentium III 600MHz	D9128A	HP	1	1270	5	6,350	
HP E60 Support Pack 3 years, 24x7, 4 hour response	H5512A	HP	2	988	5		4,940
128MB DRAMs for E60	D7156A	HP	1	205	20	4,100	
HP NetServer 10/100TX PCI LAN Adapter	D5013A	HP	1	69	5	345	
HP 15" VGA Monitor	D2828A	HP	1	185	5	925	
					Client Hardware Subtotal	**11,720**	**4,940**
Microsoft Windows 2000 Server	C11-0016	MS	1	790	5	3,950	
Microsoft Visual C++ Professional 6.0	716856	MS	1	449	1	449	
					Client Software Subtotal	**4,399**	**0**
HP Procurve Switch 2424M, Lifetime Warranty + 10% spares	J4093A	HP	1	940	8	7,520	
					Connectivity Subtotal	**7,520**	**0**
					Total	**$385,370**	**$49,668**

Notes: Price key: 1 = Software House, 2 = HP Corporate Price list

5-yr Cost of Ownership:	**$435,038**
tpmC:	**43,046.55**
$/tpmC:	**$10.11**

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

Numerical Quantities Summary for HP NetServer LXr 8500DC

MQTH, Computed Maximum Qualified Throughput **43046.55 tpmC**

Response Times (in seconds)

	90th %-ile	Maximum	Average
New-Order	0.47s	5.41s	0.32s
Payment	0.37s	5.28s	0.22s
Order-Status	0.39s	5.29s	0.25s
Delivery (interactive portion)	0.13s	0.62s	0.11s
Delivery (deferred portion)	0.61s	2.63s	0.38s
Stock-Level	1.91s	5.10s	1.01s
Menu	0.13s	0.76s	0.11s

Response time delay added for emulated components 0.1 seconds

Transaction Mix, in percent of total transactions

New-Order	44.82%
Payment	43.10%
Delivery	4.06%
Stock-Level	4.01%
Order-Status	4.01%

Keying/Think Times

	Keying Time			Think Time		
	Min	Avg	Max	Min	Avg	Max
New-Order	18s	18.01s	18.03s	0s	12.05s	120.5s
Payment	3s	3.01s	3.03s	0s	12.03s	120.5s
Order-Status	2s	2.01s	2.03s	0s	10.01s	100.6s
Delivery (interactive)	2s	2.01s	2.03s	0s	5.05s	50.4s
Stock-Level	2s	2.01s	2.03s	0s	5.06s	50.4s

Test Duration

Ramp up time	20 minutes
Measurement interval	20 minutes
Transactions during measurement interval	1998965
Ramp down time	47 minutes

Checkpointing

Number of checkpoints in measurement interval	1
Checkpoint Interval	20 minutes

Reproducibility Run

Throughput	42913.95 tpmC
Relative to MQTH	-0.31%

Table of Contents

Preface

Document Structure

This is the full disclosure report for a benchmark test of the HP NetServer LXr 8500 using Microsoft SQLServer 2000 Enterprise Edition. It meets the requirements of the TPC Benchmark ® C Standard Specification, Revision 5.0 dated February 26th, 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 NetServer LXr 8500. The full configuration was built by adding additional memory, additional disk adapters and drives. The operating system used on the server was Microsoft Windows 2000 Advanced Server and the database was Microsoft SQLServer 2000 Enterprise Edition.

The processor architecture of the HP NetServer LXr 8500 was designed for the Intel Pentium III Xeon processor. The HP NetServer LXr 8500 used in this test was powered by eight 700MHz Intel Pentium III Xeon processors, each with 2MB of 2nd level cache.

This configuration used 8 GB of HP RAM. This was achieved by using sixteen 512 MB DIMMs spread across the two memory boards.

The operating system, all executables and libraries, the master database, and swap space were contained in one 9GB hard disk, attached to an embedded PCI SCSI controller.

This measured configuration used seven HP NetRAID 4M RAID Disk Array Controllers (DACs), which were plugged into seven PCI slots on the motherboard. The database log drives consisted of eight pairs of 18GB 10krpm Ultra Wide hard disks attached to one DAC. The TPC-C database storage consisted of 240 HP 9.1GB 10krpm hard drives and 36 HP 18GB 10krpm hard drives. The 9.1GB drives on each DAC were equally distributed across the 4 SCSI channels. The 36 18GB drives were distributed across 3 SCSI channels. The channels were striped using the HP NetRAID 4M RAID configuration utility. Controller write-back caching and read ahead were specifically disabled for the PCI DACs.

Each of the five clients is a HP NetServer E60 with a single Pentium III 600MHz, 512 MB RAM, one 9.1GB SCSI hard disk, running Microsoft Windows 2000 Server.

The server, clients and the simulated users were networked together via four HP Procurve 2424M 10/100base T switches. Seven remote terminal emulators (RTEs) emulated 34,300 users executing the standard TPC-C workload. The switches are connected to each client machine through 100base T links and to all seven RTEs at 10Mbit/sec. Each client had two LAN adapters, one of which was embedded. On each client, one of the LAN adapters was connected to the server through a 100base T link and the other adapter was connected via another switch to seven RTEs at 10Mbit/sec, half-duplex. Five clients driven through seven network segments each, provided 35 network segments and 980 emulated users per network segment for the 34,300 emulated users.

HP VGA displays were used on the server and each of the clients.

General Items

Test Sponsor

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

The Enterprise NetServer Division of the 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 client machines running the Microsoft Internet Information Server (IIS) web server. The client machines used the COM+ transaction monitor (TM) to communicate with the database server.

On each 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 DBLIB 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 DBLIB 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 180 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 configurations are shown in Figures 1 and 2.

Figure 1. Measured Configuration

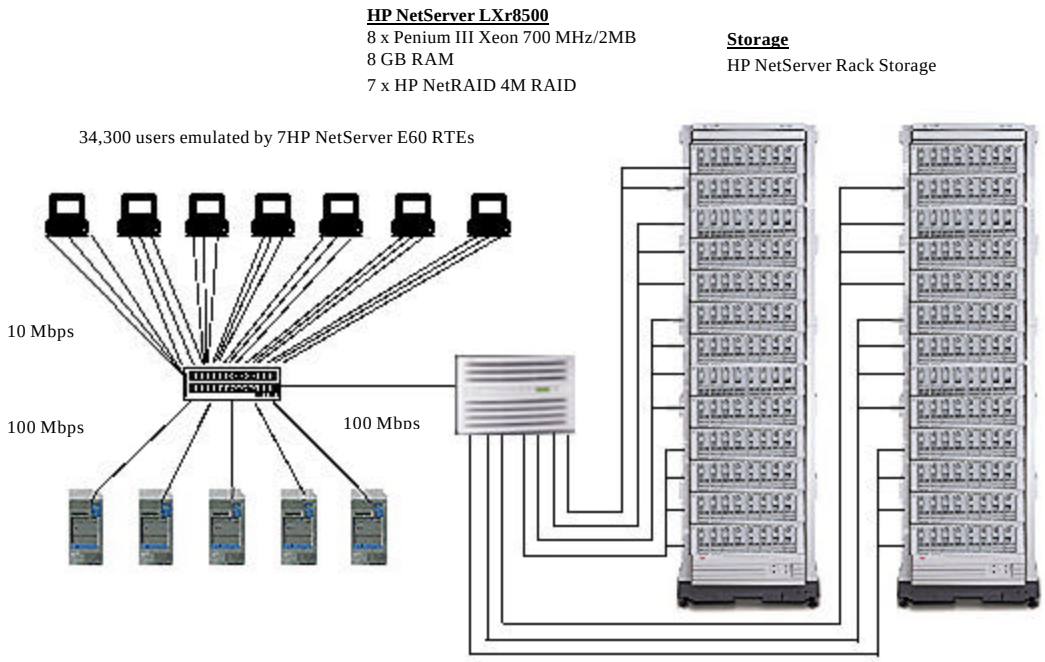
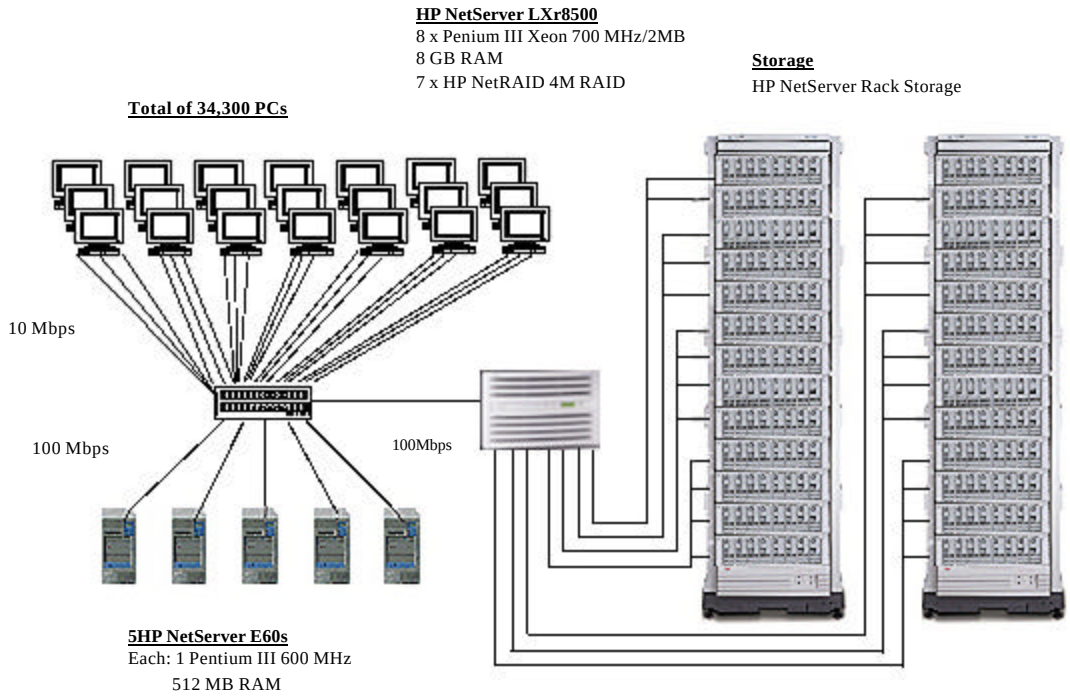


Figure 2. Priced Configuration



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 293 disks, which included 240 9GB Hot Swap disk drives and 36 18GB Hot Swap disk drives for data, 16 18GB drives for logs, and one 9GB drive for the operating system.

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	0.99%
	Average items per order	10.00
Payment	Home warehouse	85.02%
	Remote warehouse	14.98%
	Non primary key access	59.99%
Order Status	Non primary key access	59.94%
Delivery	Skipped transactions	0
Transaction Mix	New Order	44.82%
	Payment	43.10%
	Delivery	4.06%
	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 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 DBLIB 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 NetServer LXr 8500. A fully scaled database was used except for the durability tests of durable media failure. The test was performed on a database scaled to 10 warehouses, using the standard driving mechanism. However a fully scaled database under a full load would also pass this durability test.

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.

A fully scaled database was used for the Loss of System/Memory test while a 10 warehouse database was used for the Loss of Data and Loss of Log tests. With this exception of scaling, all other aspects of the configurations on the 10 warehouse database were identical to the fully scaled database configuration, including the use of the standard RTE drivers. Given this, the Loss of Data and Loss of Log tests would pass in a fully scaled database configuration.

3.5.1 Loss of Data / Loss of Log

Loss of data was demonstrated on a 10 warehouse database for convenience. The standard driving mechanism was used to generate the transaction load of 100 users for the test. To demonstrate recovery from a permanent failure of durable media containing TPC-C tables, the following steps were executed:

1. A 10 warehouse database was built having similar characteristics to fully scaled database.
2. The database was backed up using SQLServer backup facilities.
3. A sum of D_NEXT_O_ID was taken.
4. 100 users were logged in to the database and ran transactions.
5. One (mirrored) log disk drive was removed with no effect on NT or SQLServer.
6. After 5 minutes, one data disk drive was removed causing SQLServer errors.
7. The RTE was allowed to continue running. Completed transactions enroute from the clients were recorded. Error messages started appearing on the RTE screen.
8. All users were paused and stopped from the RTE.
9. SQLServer was stopped and restarted and a dump of the transaction log was taken.
10. SQLServer was stopped, Windows 2000 was shutdown, and the machine was powered off.
11. The failed disks were replaced and the controller configuration utility was run to make the two disks 'online'.
12. The machine was powered up, Windows 2000 and SQLServer were started.
13. The TPC-C database was dropped and restored from the backup.
14. The transaction log was restored and transactions rolled forward.
15. A new count of D_NEXT_O_ID was taken.
16. This number was compared with the number of new orders reported by the RTE.

3.5.2 Loss of System / Memory

This was demonstrated on the full database with 3780 warehouses in a single test. The standard driving mechanism was used to generate the transaction load of 37,800 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. 34,300 users were logged in to the database and ran transactions.
4. The system ran for 5 minutes after all the users were activated. Then a checkpoint was issued.
5. 2 minutes after the checkpoint completed, the server machine was powered 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

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 have one embedded dual-channel Ultra-2 SCSI PCI Disk controllers and 7 HP NetRAID 4M 4-channel PCI Disk Array Controllers (DACs). These cards plugged into PCI slots on the motherboard.

One hard disk was attached to the embedded PCI SCSI controllers. This drive was used for the operating system.

For the measured configuration, the database was built with 293 disks – 240 9GB for data, 36 18GB for data, 16 18GB for log, and 1 9GB for the OS and application software. The priced configuration is identical to the measured one. The data drives were all configured as hardware RAID 0. Logs were configured as hardware RAID 10. DACs 1, 2, 5, 6, and 7 were configured with 12 disk drives per RAID stripe and 4 spans for a total of 48 disk drives per data controller. DAC 4 contained 36 18 GB drives. DAC 3 was configured with 2 mirrored drives per RAID stripe and a 8 drive span for a total of 16 log drives. DAC 1, 2, 5 and 6 contained 3 partitions: partition 1 for customer/stock, partition 2 for miscellaneous, and partition 3 for backup. The other two data controllers didn't have the space for backup. Write caching was disabled on both the controller and on all the physical data drives. Write caching was enabled on the log drives. For the priced configuration, these log disks were backed up by a UPS.

Table 2 shows the complete data distribution.

Table 2: Data Distribution

HP NETRAID4M CONFIGURATION					WIN2K DISK ADMINISTRATION		
Controller #1					Disk 1: 60.0 GB		Disk 2: 173.37 GB
SCSI ID	Channels				Partitions (RAID 0)		Partition (RAID 10)
	0	1	2	3	1	2	1
0	A0-1	A1-1	A2-1	A3-1	F: CS1 Unknown 39.06 GB	O: MISC1 Unknown 20.94 GB	W: BACKUP NTFS 173.37 GB
1	A0-2	A1-2	A2-2	A3-2			
2	A0-3	A1-3	A2-3	A3-3			
3	A0-4	A1-4	A2-4	A3-4			
8	A0-5	A1-5	A2-5	A3-5			
9	A0-6	A1-6	A2-6	A3-6			
10	A0-7	A1-7	A2-7	A3-7			
11	A0-8	A1-8	A2-8	A3-8			
12	A0-9	A1-9	A2-9	A3-9			
13	A0-10	A1-10	A2-10	A3-10			
14	A0-11	A1-11	A2-11	A3-11			
15	A0-12	A1-12	A2-12	A3-12			

HP NETRAID4M CONFIGURATION					WIN2K DISK ADMINISTRATION		
Controller #2					Disk 3: 60.0 GB		Disk 4: 173.37 GB
SCSI ID	Channels				Partitions (RAID 0)		Partition (RAID 10)
	0	1	2	3	1	2	1
0	A0-1	A1-1	A2-1	A3-1	G: CS1 Unknown 39.06 GB	P: MISC1 Unknown 20.94 GB	V: BACKUP NTFS 173.37 GB
1	A0-2	A1-2	A2-2	A3-2			
2	A0-3	A1-3	A2-3	A3-3			
3	A0-4	A1-4	A2-4	A3-4			
8	A0-5	A1-5	A2-5	A3-5			
9	A0-6	A1-6	A2-6	A3-6			
10	A0-7	A1-7	A2-7	A3-7			
11	A0-8	A1-8	A2-8	A3-8			
12	A0-9	A1-9	A2-9	A3-9			
13	A0-10	A1-10	A2-10	A3-10			
14	A0-11	A1-11	A2-11	A3-11			
15	A0-12	A1-12	A2-12	A3-12			

HP NETRAID4M CONFIGURATION					WIN2K DISK ADMINISTRATION
Controller #3					Disk 5: 135.59 GB
SCSI ID	Channels				Partition (RAID 10)
	0	1	2	3	1
0	A0-1	A1-1			E: LOG Unknown 135.59 GB
1	A0-2	A1-2			
2	A0-3	A1-3			
3	A0-4	A1-4			
8	A0-5	A1-5			
9	A0-6	A1-6			
10	A0-7	A1-7			
11	A0-8	A1-8			
12					
13					
14					
15					

HP NETRAID4M CONFIGURATION					WIN2K DISK ADMINISTRATION		
Controller #4					Disk 6: 610.16 GB		
SCSI ID	Channels				Partitions (RAID 0)		
	0	1	2	3	1	2	3
0	A0-1	A1-1	A2-1	A3-1	H: CS3 Unknown 54.69 GB	P: MISC3 Unknown 26.37 GB	Freespace 529.11 GB
1	A0-2	A1-2	A2-2	A3-2			
2	A0-3	A1-3	A2-3	A3-3			
3	A0-4	A1-4	A2-4	A3-4			
8	A0-5	A1-5	A2-5	A3-5			
9	A0-6	A1-6	A2-6	A3-6			
10	A0-7	A1-7	A2-7	A3-7			
11	A0-8	A1-8	A2-8	A3-8			
12	A0-9	A1-9	A2-9	A3-9			
13	A0-10	A1-10	A2-10	A3-10			
14	A0-11	A1-11	A2-11	A3-11			
15	A0-12	A1-12	A2-12	A3-12			

HP NETRAID4M CONFIGURATION					WIN2K DISK ADMINISTRATION		
Controller #5					Disk 7: 60.0 GB		Disk 8: 173.37 GB
SCSI ID	Channels				Partitions (RAID 0)		Partition (RAID 10)
	0	1	2	3	1	2	1
0	A0-1	A1-1	A2-1	A3-1	I: CS1 Unknown 39.06 GB	R: MISC1 Unknown 20.94 GB	X: BACKUP NTFS 173.37 GB
1	A0-2	A1-2	A2-2	A3-2			
2	A0-3	A1-3	A2-3	A3-3			
3	A0-4	A1-4	A2-4	A3-4			
8	A0-5	A1-5	A2-5	A3-5			
9	A0-6	A1-6	A2-6	A3-6			
10	A0-7	A1-7	A2-7	A3-7			
11	A0-8	A1-8	A2-8	A3-8			
12	A0-9	A1-9	A2-9	A3-9			
13	A0-10	A1-10	A2-10	A3-10			
14	A0-11	A1-11	A2-11	A3-11			
15	A0-12	A1-12	A2-12	A3-12			

HP NETRAID4M CONFIGURATION					WIN2K DISK ADMINISTRATION		
Controller #6					Disk 9: 60.0 GB		Disk 10: 173.37 GB
SCSI ID	Channels				Partitions (RAID 0)		Partition (RAID 10)
	0	1	2	3	1	2	1
0	A0-1	A1-1	A2-1	A3-1	J: CS1 Unknown 39.06 GB	S: MISC1 Unknown 20.94 GB	Y: BACKUP NTFS 173.37 GB
1	A0-2	A1-2	A2-2	A3-2			
2	A0-3	A1-3	A2-3	A3-3			
3	A0-4	A1-4	A2-4	A3-4			
8	A0-5	A1-5	A2-5	A3-5			
9	A0-6	A1-6	A2-6	A3-6			
10	A0-7	A1-7	A2-7	A3-7			
11	A0-8	A1-8	A2-8	A3-8			
12	A0-9	A1-9	A2-9	A3-9			
13	A0-10	A1-10	A2-10	A3-10			
14	A0-11	A1-11	A2-11	A3-11			
15	A0-12	A1-12	A2-12	A3-12			

HP NETRAID4M CONFIGURATION					WIN2K DISK ADMINISTRATION		
Controller #7					Disk 11: 406.75 GB		
SCSI ID	Channels				Partitions (RAID 0)		
	0	1	2	3	1	2	3
0	A0-1	A1-1	A2-1	A3-1	K: CS3 Unknown 54.69 GB	T: MISC3 Unknown 26.37 GB	Freespace 325.69 GB
1	A0-2	A1-2	A2-2	A3-2			
2	A0-3	A1-3	A2-3	A3-3			
3	A0-4	A1-4	A2-4	A3-4			
8	A0-5	A1-5	A2-5	A3-5			
9	A0-6	A1-6	A2-6	A3-6			
10	A0-7	A1-7	A2-7	A3-7			
11	A0-8	A1-8	A2-8	A3-8			
12	A0-9	A1-9	A2-9	A3-9			
13	A0-10	A1-10	A2-10	A3-10			
14	A0-11	A1-11	A2-11	A3-11			
15	A0-12	A1-12	A2-12	A3-12			

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	3,430
District	34,300
Customer	102,900,000
History	102,900,000
Orders	102,900,000
New Orders	30,870,000
Order Line	468,001,833
Stock	100,000
Item	343,000,000

No rows were deleted for the benchmark runs.

4.3 180 Day Space

Details of the 180 day space computations along with proof that the database is configured to sustain 8 hours of growth for the dynamic tables 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 100.92 GB (including mirror) to sustain the log for 8 hours. Space available for the transaction log was 135.59 GB (including mirror), 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 180 day 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 SQLServer 2000 Enterprise Edition 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_cs_fg and MSSQL70_misc_fg. MSSQL70_cs_fg consist of 6 partitions at 34 GB each and MSSQL70_misc_fg consist of 6 partitions at 16 GB each as shown in the createdb.sql. The log was configured with 120 GB.

Chapter 5 Performance Metrics and Response Time

5.1 Throughput

Measured tpmC® must be reported.

Measured TpmC®: 43,046.55
Price per TpmC®: 10.11

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	0.32s	0.47s	5.41s
Payment	0.22s	0.37s	5.28s
Order-Status	0.25s	0.39s	5.29s
Delivery (interactive portion)	0.11s	0.13s	0.62s
Delivery (deferred portion)	0.38s	0.61s	2.63s
Stock-Level	1.01s	1.91s	5.10s
Menu	0.11s	0.13s	0.76s

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	18.01s	18.03s
Payment	3s	3.01s	3.03s
Order Status	2s	2.01s	2.03s
Interactive Delivery	2s	2.01s	2.03s
Stock Level	2s	2.01s	2.03s

Table 6: Transaction Think Times

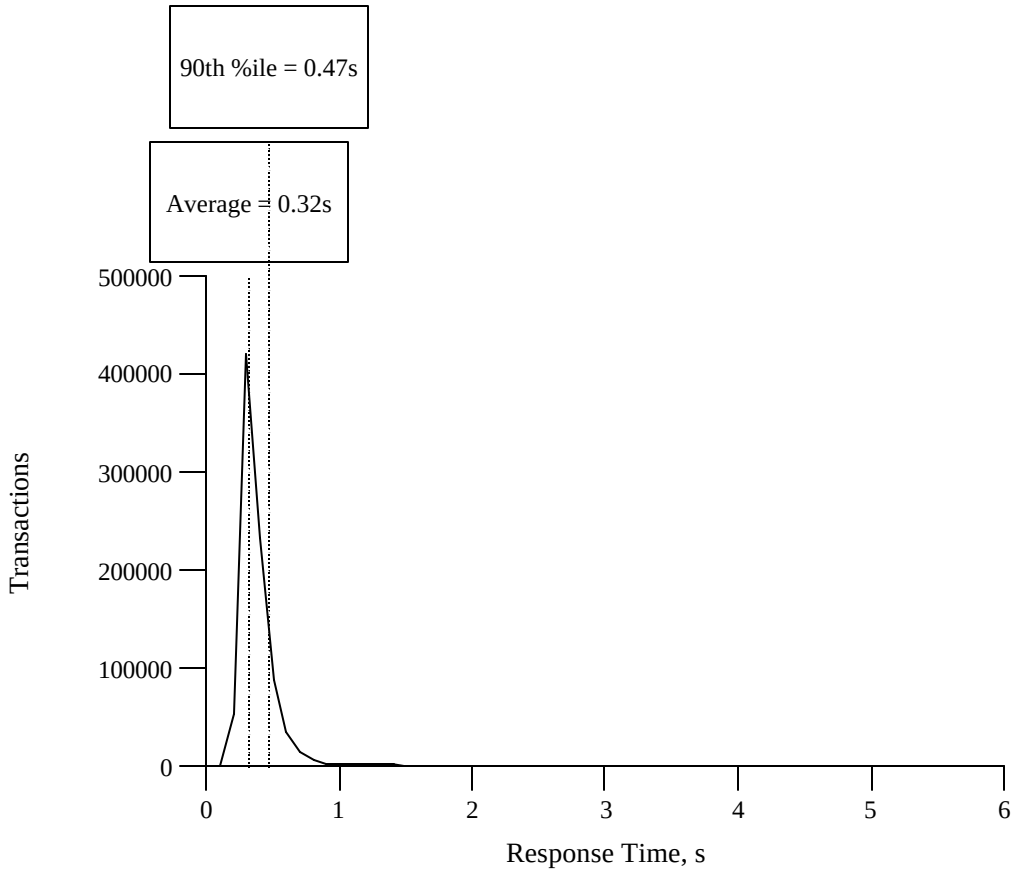
Think Times	Minimum	Average	Maximum
New Order	0s	12.05s	120.5s
Payment	0s	12.03s	120.5s
Order Status	0s	10.01s	100.6s
Interactive Delivery	0s	5.05s	50.4s
Stock Level	0s	5.06s	50.4s

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

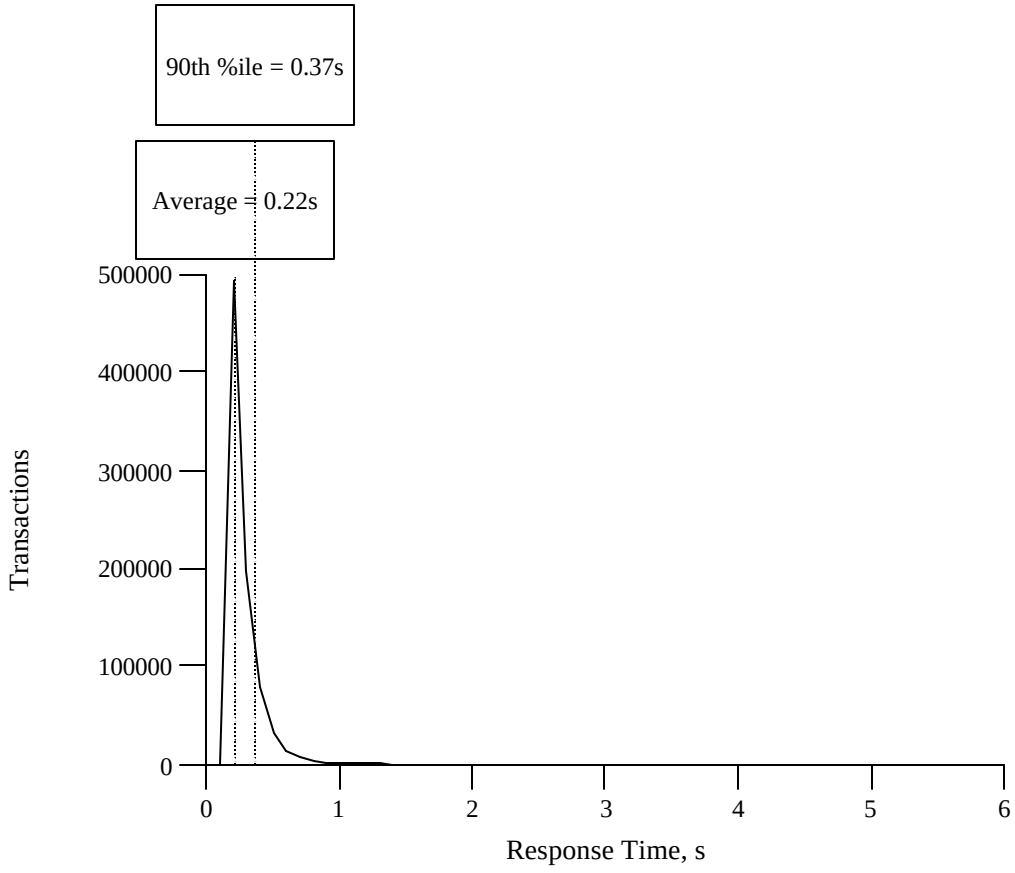
Figure 3: New Order Response Time Distribution



Response time frequency distribution for New Order transaction

5.4.2 Payment Response Time Distribution

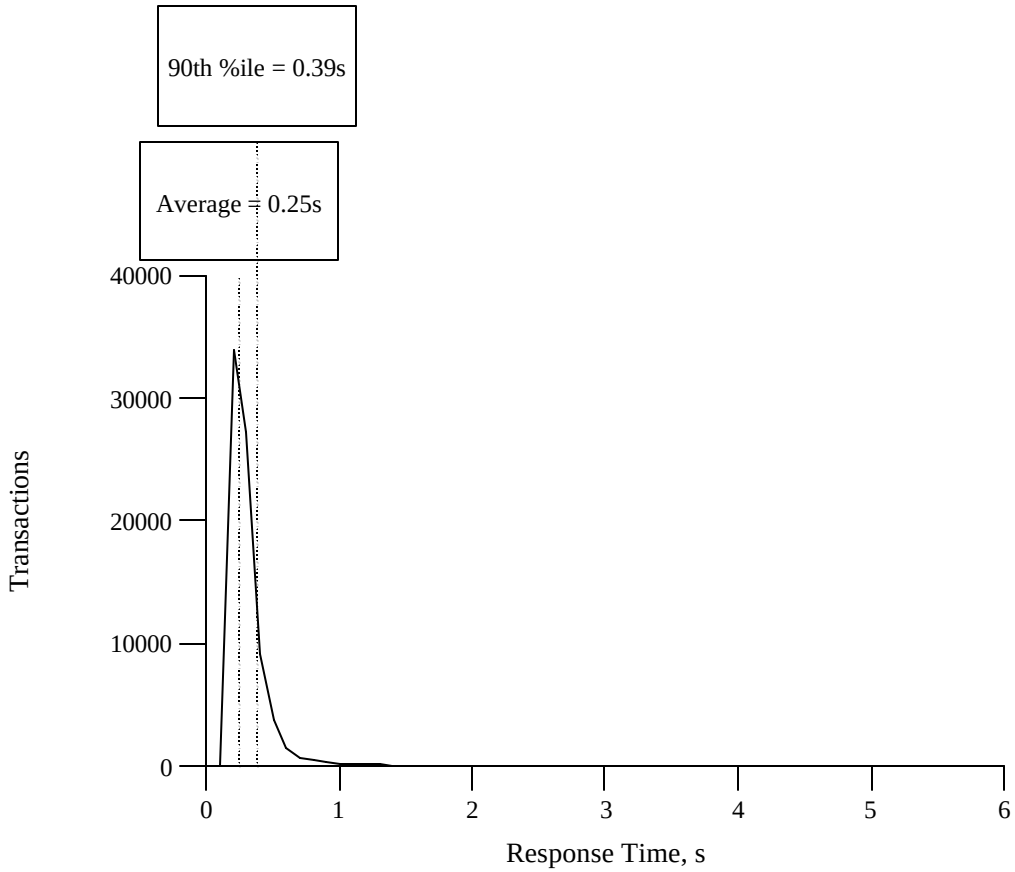
Figure 4: Payment Response Time Distribution



Response time frequency distribution for Payment transaction

5.4.3 Order Status Response Time

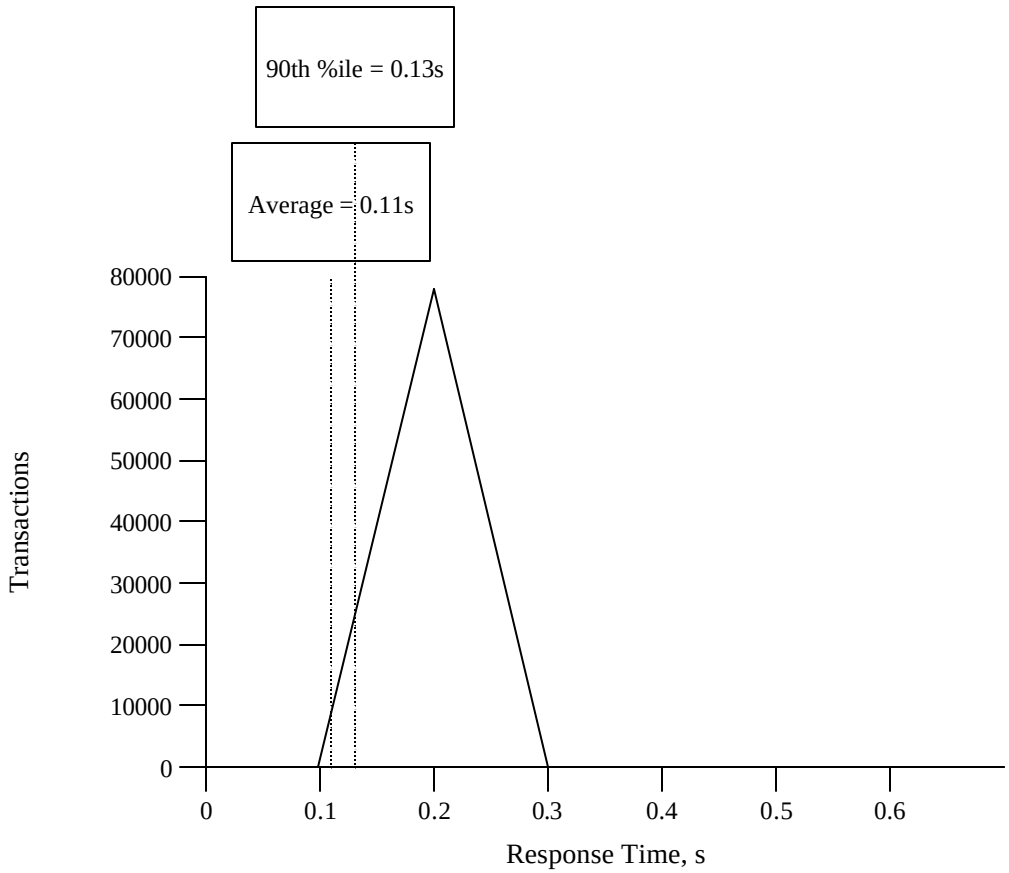
Figure 5: Order Status Response Time Distribution



Response time frequency distribution for Order Status transaction

5.4.4 Delivery Response Time Distribution

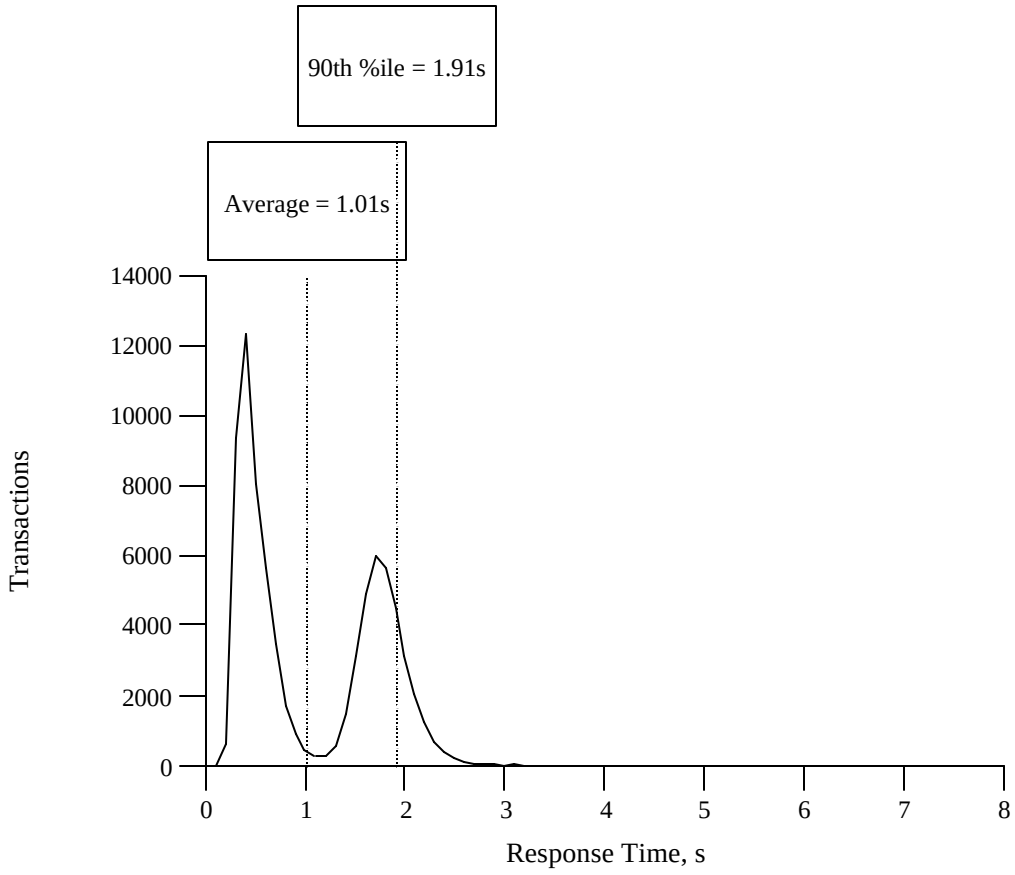
Figure 6: Delivery Response Time Distribution



Response time frequency distribution for Delivery transaction

5.4.5 Stock Level Response Time

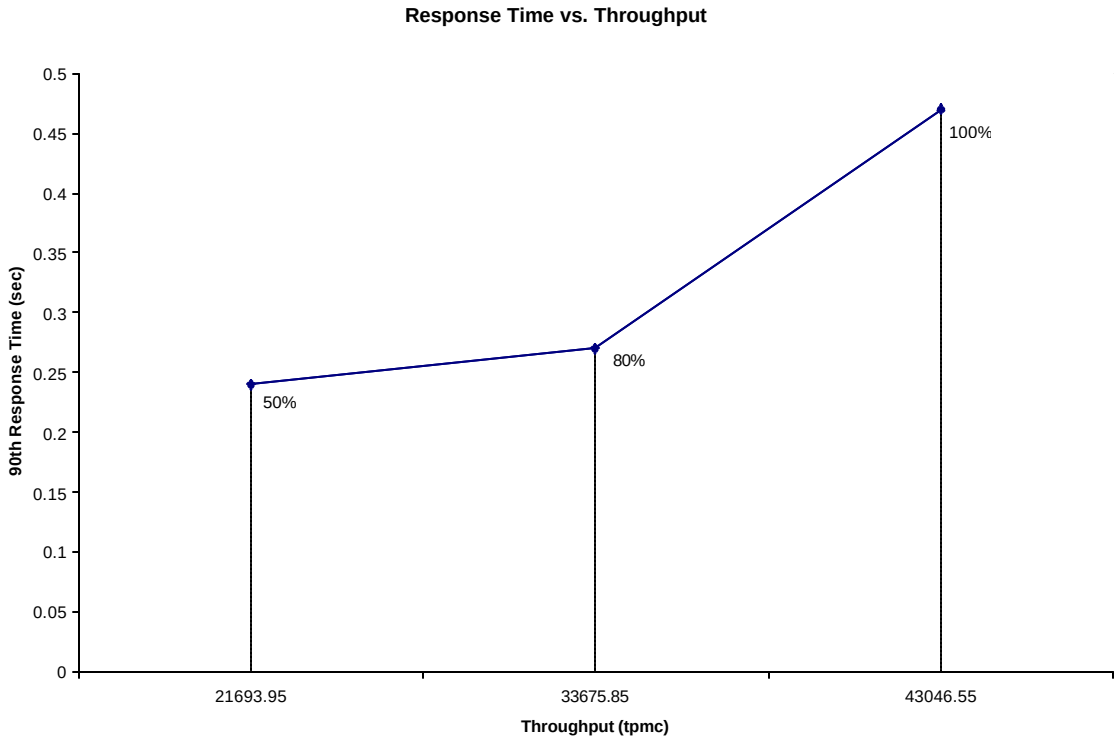
Figure 7: Stock Level Response Time Distribution



Response time frequency distribution for Stock Level transaction

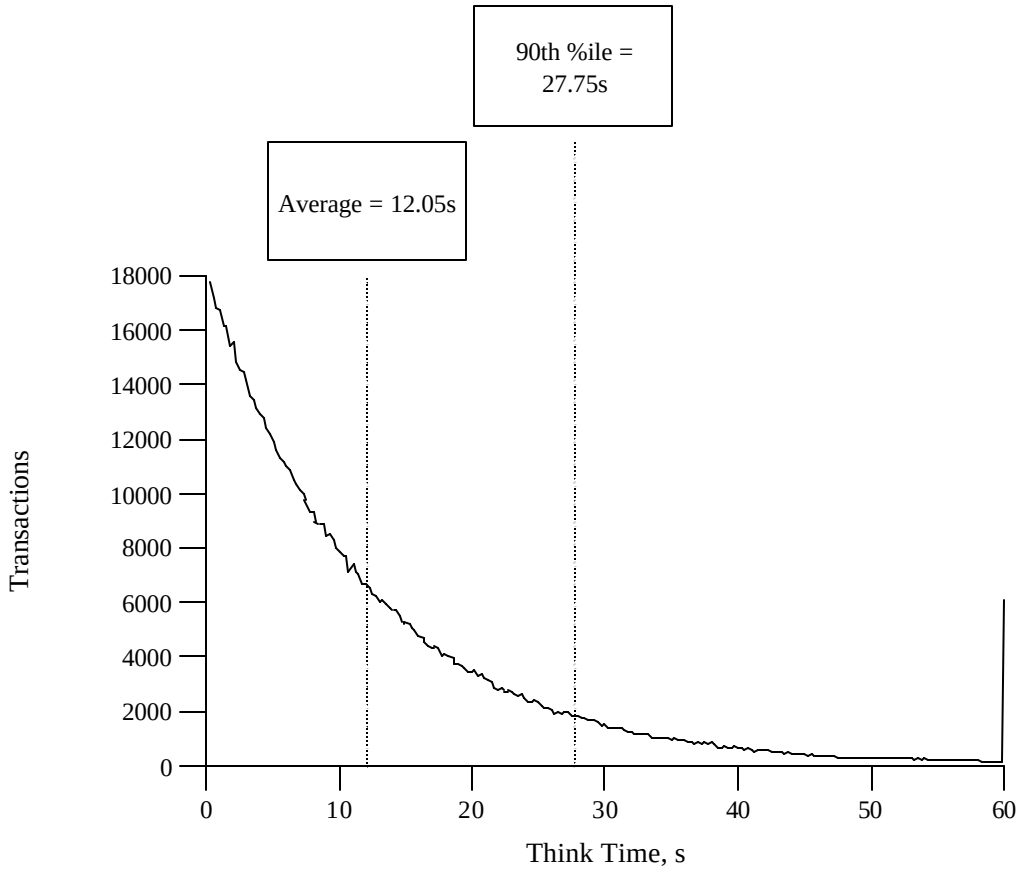
5.4.6 Response Time Versus Throughput

Figure 8: New Order Response Time Distribution



5.4.7 New Order Think Time Distribution

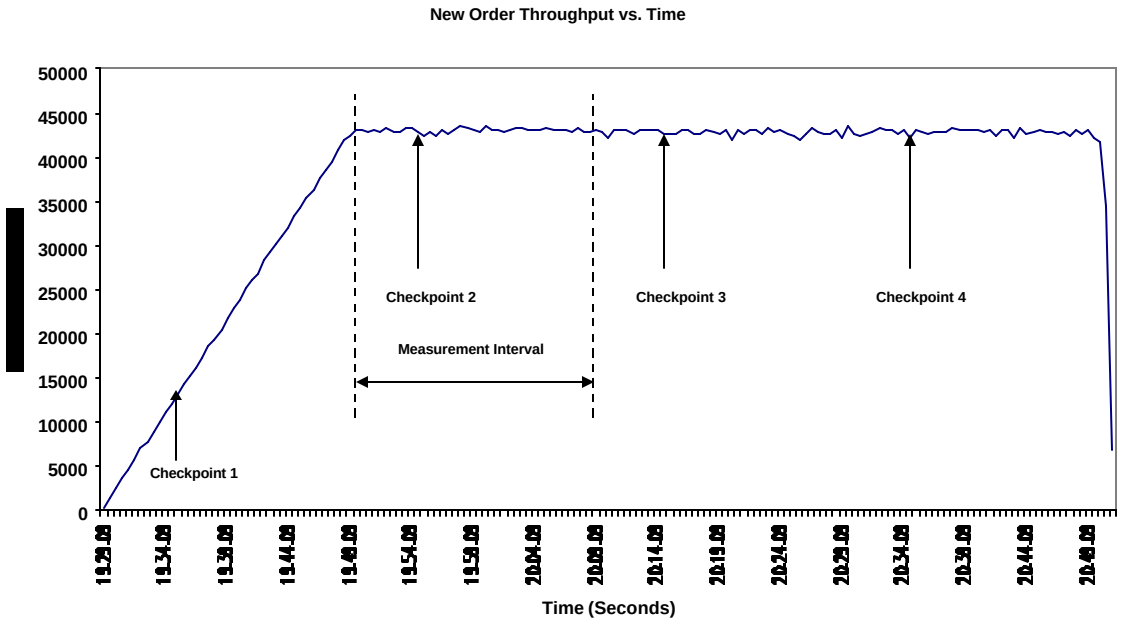
Figure 9: New Order Think Time Distribution



Think time frequency distribution for New Order transaction

5.4.8 Throughput Versus Time Distribution

Figure 10: New Order Throughput versus Time



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 24 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 6 minutes. 2 manual checkpoints were invoked during login phase to make sure that the automatic checkpoint would never get to execute. The interval between each pair was as short as 10 minutes. During the ramp -up and after all users were active, a background process slept and performed the checkpoint every 20 minutes. The measurement interval was equal to 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 and is depicted on the graph in Figure 10.

5.7 Reproducibility

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

A second measurement achieved a throughput of 42,913.95 tpmC® during a 20-minute, steady state interval.

5.8 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 20 minutes.

5.9 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.

5.10 Transaction Mix

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

Table 7: Transaction Mix

Type	Percentage
New Order	44.82%
Payment	43.10%
Delivery	4.06%
Stock Level	4.01%
Order Status	4.01%

5.11 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.

5.12 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 20 minutes. There is one checkpoint within the measurement interval. This checkpoint starts 378 seconds into the measurement interval. The checkpoint interval (time between starts of two consecutive checkpoints) is also 20 minutes. Each checkpoint took approximately 360 seconds to complete. In conformance with Clause 5.5.2.2, the checkpoint occurs outside the guard zones.

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 replacing the workstations and hubs connected via LANs.

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

The bandwidth of the measured and priced configurations were:
- 10base T (10Mbit/sec) network segments between RTEs and the switches.
- 100base T (100Mbit/sec) network segments between the clients and the 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 3 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 3-year pricing, price/performance and the availability date must be included.

Table 8: Throughput, Price Performance and Availability

Maximum qualified throughput:	43046.55 tpmC
Price per tpmC:	$10.11
Availability:	January 9, 2001

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:
- 1 Microsoft SQLServer 2000 Enterprise Edition license.
- 1 Microsoft Windows 2000 Advanced Server license.
- 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:

Performance Metrics, Inc.
137 Yankton St., Suite 101
Folsom, CA 95630
U.S.A.
Phone: 916 985-1131
Fax: 916 985-1185

The auditor was Tom Sawyer.

The upgrade to the TPC-C Revision 5.0 specification was audited by:

Performance Metrics, Inc.
137 Yankton St., Suite 101
Folsom, CA 95630
U.S.A.
Phone: 916 985-1131
Fax: 916 985-1185

The auditor was Lorna Livingtree.

Requests for this Full Disclosure Report (FDR) should sent to:

Hewlett-Packard Company
Enterprise NetServer Division
10955 Tantau Avenue
Cupertino, CA 95014-0770 USA
Attn: Alexander Carlton, MS 45NUH

A copy of the attestation letters received from the auditors follow:

PERFORMANCE METRICS INC.
TPC Certified Auditors

December 28, 2000

Mr. Dave Tanis
Project Manager, Performance Engineering
Network Server Division
Hewlett-Packard Company
10955 Tantau Avenue
Cupertino, CA 95014

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

Platform:	Hewlett-Packard NetServer LXr8500
Database Manager:	Microsoft SQL Server 2000 Enterprise Edition
Operating System:	Microsoft Windows 2000 Advanced Server
Transaction Manager:	Microsoft Com+

Server: Hewlett-Packard NetServer LXr8500				
CPU's	Memory	Disks	90% Response	TpmC
8 Pentium III Xeon @ 700 MHz	Main: 8 GB Cache: 2MB each	241 @ 9GB 52 @ 18GB	**0.47 sec.**	**43,046.55**
5 Clients: Hewlett-Packard NetServer E60				
1 Pentium III @ 600 MHz	Main: 512 MB Cache: 512 KB	1 @ 9GB	Na	Na

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

- The transactions were correctly implemented.
- The database files were properly sized and populated.

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

- The database was properly scaled with 3,430 warehouses.
- There were 34,300 emulated users present for the measurement
- The ACID properties were met.
- The Durability tests were performed on the measured database except the loss-of-data-disk test which was performed on a 10-warehouse database.
- Input data was generated according to the specified percentages.
- Eight hours of durable log space was present on the tested system.
- Space for eight hours of growth in dynamic tables was present on the tested system.
- The data for the 180-day space calculation was verified – the measured database had sufficient space.
- The steady state portion of the test was 20 minutes.
- One checkpoint was taken before the measured interval.
- One checkpoint was taken during the measured interval.
- The checkpoints were verified to be clear of the guard zone.
- The system pricing was checked for major components and maintenance.

Auditor Notes:

None.

Sincerely,

Tom Sawyer
Auditor

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

PERFORMANCE METRICS INC.
TPC Certified Auditors

August 22, 2001

Alex Carlton
Hewlett-Packard Network Server Division
10955 Tantau Ave. MS 45NU-H
Cupertino, CA 95014

In my opinion, the data provided for the HP LXr8500 8way results of January 10, 2001 complies with the TPC-C Version 5 upgrade requirements.

The following attributes of the benchmark were given special attention:

- The data for the 60 day space calculation was verified
- Maintenance was verified to be 3-year, 7 X 24 with 4 hour response time.
- The system pricing was checked for major components and maintenance.

Auditor Notes:

none

Sincerely,

Lorna Livingtree

Lorna Livingtree
Auditor

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

Appendix A Application Source

A.1 Client Front End

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
      TOPMARGIN, 7
      BOTTOMMARGIN, 88
   END
END
#endif    // APSTUDIO_INVOKED

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



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

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
```

APPENDIX A - APPLICATION SOURCE CODE

```
#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 );
```

APPENDIX A - APPLICATION SOURCE CODE

```
                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);
```

APPENDIX A - APPLICATION SOURCE CODE

```
                                        // 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 exp ected version
number.
 *
 * RETURNS:                             TRUE            inet service expected return value.
 */

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

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

```
		return TRUE;
}

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

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

		TermDeleteAll();
		return TRUE;
}


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

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

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

#ifdef ICECAP
		StartCAP();
#endif

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

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

					throw new CWEBCLNT_ERR( ERR_INVALID_TERMID );
				}

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

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


			switch(iCmd)
			{
			case 0:
				WelcomeForm(pECB, szBuffer);
				break;
			case 1:
				switch( FormId )
				{
					case WELCOME_FORM:
					case MAIN_MENU_FORM:
						break;
					case NEW_ORDER_FORM:
						ProcessNewOrderForm(pECB, TermId, szBuffer);
						break;
					case PAYMENT_FORM:
						ProcessPaymentForm(pECB, TermId, szBuffer);
						break;
					case DELIVERY_FORM:
						ProcessDeliveryForm(pECB, TermId, szBuffer);
						break;
					case ORDER_STATUS_FORM:
						ProcessOrderStatusForm(pECB, TermId, szBuffer);
						break;
					case STOCK_LEVEL_FORM:
```

```
                                        ProcessStockLevelForm(pECB, TermId, szBuffer);
                                        break;
                        }
                        break;
                case 2:
                        // new-order selected from menu; display new-order input form
                        MakeNewOrderForm(TermId, NULL, INPUT_FORM, szBuffer);
                        break;
                case 3:
                        // payment selected from menu; display payment input form
                        MakePaymentForm(TermId, NULL, INPUT_FORM, szBuffer);
                        break;
                case 4:
                        // delivery selected from menu; display  delivery input form
                        MakeDeliveryForm(TermId, NULL, INPUT_FORM, szBuffer);
                        break;
                case 5:
                        // order-status selected from menu; display order-status input form
                        MakeOrderStatusForm(TermId, NULL, INPUT_FORM, szBuffer);
                        break;
                case 6:
                        // stock-level selected from menu; display stock-level input form
                        MakeStockLevelForm(TermId, NULL, INPUT_FORM, szBuffer);
                        break;
                case 7:
                        // ExitCmd
                        TermDelete(TermId);
                        WelcomeForm(pECB, szBuffer);
                        break;
                case 8:
                        SubmitCmd(pECB, szBuffer);
                        break;
                case 9:
                        // menu
                        MakeMainMenuForm(TermId, Term.pClientData[TermId].iSyncId, szBuffer);
                        break;
                case 10:
                        // CMD=Clear
                        // resets all connections; should only be used when no other connections are active
                        TermDeleteAll();
                        TermInit();
                        WelcomeForm(pECB, szBuffer);
                        break;
                case 11:        // CMD=Stats
                        StatsCmd(pECB, szBuffer);
                        break;
                }
        }
        catch (CBaseErr *e)
        {
                ErrorForm( pECB, e->ErrorType(), e->ErrorNum(), TermId, iSyncId, e->ErrorText(), szBuffer );
                delete e;
        }
        catch (...)
        {
                ErrorForm( pECB, ERR_TYPE_WEBDLL, 0, TermId, iSyncId, "Error: Unhandled exception in Web
Client.", szBuffer );
                }

        #ifdef ICECAP
                StopCAP();
        #endif


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

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

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


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

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

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

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

              (VOID) DeregisterEventSource(hEventSource);
           }
        }
```

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

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

        DELIVERY_TRANSACTION                    delivery;
        PDELIVERY_DATA                          pDeliveryData;
        TXN_RECORD_TPCC_DELIV_DEF       txnDeliRec;

        DWORD                           index;
        HANDLE                          handles[2];

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

        assert(txnDelilog != NULL);

        try
        {
                if (Reg.eDB_Protocol == ODBC)
                        pTxn = pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
szMyComputerName, Reg.szDbName );
                else if (Reg.eDB_Protocol == DBLIB)
                        pTxn = pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
szMyComputerName, Reg.szDbName );
                pDeliveryData = pTxn->BuffAddr_Delivery();
        }
        catch (CBaseErr *e)
        {
                char szTmp[1024];
                wsprintf( szTmp, "Error in Delivery Txn thread.  Could not connect to database.  "
                        "%s.  Server=%s, User=%s, Password=%s, Database=%s",
                        e->ErrorText(), Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
Reg.szDbName );
                WriteMessageToEventLog( szTmp );
                delete e;
                goto ErrorExit;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception caught in DeliveryWorkerThread."));
                goto ErrorExit;
        }

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

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

                                // make a local copy of current entry from delivery buffer and increment
buffer index
                                EnterCriticalSection(&DelBuffCriticalSection);
                                delivery = *(pDelBuff+dwDelBuffBusyIndex);
                                dwDelBuffFreeCount++;
                                dwDelBuffBusyIndex++;
                                if (dwDelBuffBusyIndex == dwDelBuffSize)                // wrap-around if at
end of buffer
                                        dwDelBuffBusyIndex = 0;

                                LeaveCriticalSection(&DelBuffCriticalSection);

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

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

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

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

                                if (txnDelilog != NULL)
                                        txnDelilog->WriteToLog(&txnDeliRec);
                        }
                }
                catch (CBaseErr *e)
                {
                        char szTmp[1024];
```

```
                        wsprintf( szTmp, "Error in Delivery Txn thread.  %s", e->ErrorText() );
                        WriteMessageToEventLog( szTmp );
                        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_ER R);
			// 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.eTxn Mon == 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."
			},
```

APPENDIX A - APPLICATION SOURCE CODE

```
                { ERR_GETPROCADDR_FAILED,
        "Could not map proc in DLL.  GetProcAddr error.  DLL="          },
                { ERR_HTML_ILL_FORMED,
        "Required key field is missing from HTML string."               },
                { ERR_INVALID_SYNC_CONNECTION,          "Invalid Terminal
Sync ID."
                },
                { ERR_INVALID_TERMID,
        "Invalid Terminal ID."
                                                        },
                { ERR_LOADDLL_FAILED,
        "Load of DLL failed.  DLL="
                                                        },
                { ERR_MAX_CONNECTIONS_EXCEEDED,                 "No
connections available.  Max Connections is probably too low."  },
                { ERR_MISSING_REGISTRY_ENTRIES,         "Required registry
entries are missing.  Rerun INSTALL to correct."        },
                { ERR_NEWORDER_CUSTOMER_INVALID,                "New
Order customer id invalid data type, range = 1 to 3000."        },
                { ERR_NEWORDER_CUSTOMER_KEY,                    "New
Order missing Customer key \"CID*\""                            },
                { ERR_NEWORDER_DISTRICT_INVALID,                "New
Order District ID Invalid range 1 - 10."                        },
                { ERR_NEWORDER_FORM_MISSING_DID,                "New
Order missing District key \"DID*\""                            },
                { ERR_NEWORDER_ITEMID_INVALID,          "New Order Item Id
is wrong data type, must be numeric."                           },
                { ERR_NEWORDER_ITEMID_RANGE,                    "New
Order Item Id is out of range. Range = 1 to 999999."            },
                { ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,    "New Order Item_Id
field entered without a corresponding Supp_W."          },
                { ERR_NEWORDER_MISSING_IID_KEY,                 "New
Order missing Item Id key \"IID*\""                             },
                { ERR_NEWORDER_MISSING_QTY_KEY,                 "New
Order Missing Qty key \"Qty##*\""
        },
                { ERR_NEWORDER_MISSING_SUPPW_KEY,               "New
Order missing Supp_W key \"SP##*\""                             },
                { ERR_NEWORDER_NOITEMS_ENTERED,                 "New
Order No order lines entered."
                },
                { ERR_NEWORDER_QTY_INVALID,                     "New
Order Qty invalid must be numeric range 1 - 99."        },
                { ERR_NEWORDER_QTY_RANGE,
        "New Order Qty is out of range. Range = 1 to 99."               },
                { ERR_NEWORDER_QTY_WITHOUT_SUPPW,               "New
Order Qty field entered without a corresponding Supp_W."        },
                { ERR_NEWORDER_SUPPW_INVALID,                   "New
Order Supp_W invalid data type must be numeric."         },
                { ERR_NO_SERVER_SPECIFIED,                      "No
Server name specified."
                },
                { ERR_ORDERSTATUS_CID_AND_CLT,          "Order Status Only
Customer ID or Last Name may be entered, not both."     },
                { ERR_ORDERSTATUS_CID_INVALID,                  "Order Status
Customer ID invalid, range must be numeric 1 - 3000."           },
                { ERR_ORDERSTATUS_CLT_RANGE,                            "Order
Status Customer last name longer than 16 characters."           },
                { ERR_ORDERSTATUS_DID_INVALID,                  "Order Status
District invalid, value must be numeric 1 - 10."                },
                { ERR_ORDERSTATUS_MISSING_CID_CLT,              "Order Status Either
Customer ID or Last Name must be entered."                      },
                { ERR_ORDERSTATUS_MISSING_CID_KEY,              "Order Status
missing Customer key \"CID*\"."                                 },
                { ERR_ORDERSTATUS_MISSING_CLT_KEY,              "Order Status
missing Customer Last Name key \"CLT*\"."                       },
                { ERR_ORDERSTATUS_MISSING_DID_KEY,              "Order Status
missing District key \"DID*\"."                                 },
                { ERR_PAYMENT_CDI_INVALID,
        "Payment Customer district invalid must be numeric."            },
                { ERR_PAYMENT_CID_AND_CLT,
        "Payment Only Customer ID or Last Name may be entered, not both."       },
                { ERR_PAYMENT_CUSTOMER_INVALID,
        "Payment Customer data type invalid, must be numeric."                          },
                { ERR_PAYMENT_CWI_INVALID,
        "Payment Customer Warehouse invalid, must be numeric."                          },
                { ERR_PAYMENT_DISTRICT_INVALID,                 "Payment District ID
is invalid, must be 1 - 10."                                    },
                { ERR_PAYMENT_HAM_INVALID,
        "Payment Amount invalid data type must be numeric."                             },
                { ERR_PAYMENT_HAM_RANGE,
        "Payment Amount out of range, 0 - 9999.99."
                },
                { ERR_PAYMENT_LAST_NAME_TO_LONG,
        "Payment Customer last name longer than 16 characters."                 },
                { ERR_PAYMENT_MISSING_CDI_KEY,                  "Payment missing
Customer district key \"CDI*\"."                                },
                { ERR_PAYMENT_MISSING_CID_CLT,                  "Payment Either
Customer ID or Last Name must be entered."                      },
                { ERR_PAYMENT_MISSING_CID_KEY,                  "Payment missing
Customer Key \"CID*\"."                                         },
                { ERR_PAYMENT_MISSING_CLT_KEY,                  "Payment missing
Customer Last Name key \"CLT*\"."                               },
                { ERR_PAYMENT_MISSING_CWI_KEY,                  "Payment missing
Customer Warehouse key \"CWI*\"."                               },
                { ERR_PAYMENT_MISSING_DID_KEY,                  "Payment missing
District Key \"DID*\"."                                         },
                { ERR_PAYMENT_MISSING_HAM_KEY,                  "Payment missing
Amount key \"HAM*\"."                                           },
                { ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,  "Stock Level; missing Threshold
key \"TT*\"."                                                   },
                { ERR_STOCKLEVEL_THRESHOLD_INVALID,             "Stock Level;
Threshold value must be in the range = 1 - 99."                 },
                { ERR_STOCKLEVEL_THRESHOLD_RANGE,                       "Stock
Level Threshold out of range, range must be 1 - 99."            },
                { ERR_VERSION_MISMATCH,
        "Invalid version field.  RTE and Web Client are probably out of sync." },
```

```
		{		ERR_W_ID_INVALID,
	"Invalid Warehouse ID."
							},
		{		0,
				""

				}
	};

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

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

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

}

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

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

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

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

	*pQueryString = ptr;
	return;

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

/* FUNCTION: GetIntKeyValue
 *
 * PURPOSE:		This function parses a http formatted string for a specific key value.
 *
 * ARGUMENTS:		char			*pQueryString		http string from client browser
 *				char			*pKey
	key value to look for
 *				WEBERROR		NoKeyErr		error value to throw
if key not found
 *				WEBERROR		NotIntErr		error value to throw
if value not numeric
 *
 * RETURNS:		integer
 *
 * ERROR:		if (the pKey value is not found) then
 *					if (NoKeyErr != NO_ERR)
 *						throw CWEBCLNT_ERR(err)
 *					else
```

```
 *                                                                  return 0
 *                                                  else if (non-numeric char found) then
 *                                                          if (NotIntErr != NO_ERR) then
 *                                                                  throw CWEBCLNT_ERR(err)
 *                                                          else
 *                                                                  return 0
 *
 * COMMENTS:            http keys are formatted either KEY=value& or KEY=value\0. This DLL formats
 *                                                  TPC-C input fields in such a manner that the keys can be extracted in the
 *                                                  above manner.
 */

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

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

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

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

        *pQueryString = ptr;
        return atoi(ptr0);

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

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

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

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

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

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

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

        LeaveCriticalSection(&TermCriticalSection);
}

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

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

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

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

        LeaveCriticalSection(&TermCriticalSection);
```

```
}

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

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

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

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

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

        LeaveCriticalSection(&TermCriticalSection);
        return iNewTerm;
}

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

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

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

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

                LeaveCriticalSection(&TermCriticalSection);
        }
}


/* FUNCTION: MakeErrorForm
 */

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

/* FUNCTION: MakeMainMenuForm
 */

void MakeMainMenuForm(int iTermId, int iSyncId, char *szForm)
```

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

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

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

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

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

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

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

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

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

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

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

                strcpy( szForm+c,
                        "District: <INPUT NAME=\"DID*\" SIZE=1>                 Date:<BR>"
```

APPENDIX A - APPLICATION SOURCE CODE

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

                if ( bValid )
                {
                        c += wsprintf(szForm+c, "%2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d",
                                        pNewOrderData->o_entry_d.day,
                                        pNewOrderData->o_entry_d.month,
                                        pNewOrderData->o_entry_d.year,
                                        pNewOrderData->o_entry_d.hour,
                                        pNewOrderData->o_entry_d.minute,
                                        pNewOrderData->o_entry_d.second);
                }

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

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

                        for(i=0; i<pNewOrderData->o_ol_cnt; i++)
                        {
                                c += sprintf(szForm+c, "  %4.4d   %6.6d   %-24s  %2.2d    %3.3d    %1.1s
$%6.2f  $%7.2f  <BR>",
                                                pNewOrderData->OL[i].ol_supply_w_id,
                                                pNewOrderData->OL[i].ol_i_id,
                                                pNewOrderData->OL[i].ol_i_name,
                                                pNewOrderData->OL[i].ol_quantity,
                                                pNewOrderData->OL[i].ol_stock,
                                                pNewOrderData->OL[i].ol_brand_generic,
                                                pNewOrderData->OL[i].ol_i_price,
                                                pNewOrderData->OL[i].ol_amount );
                        }
                }
                else
                {
                        c += wsprintf(szForm+c,
                                        "%Disc:<BR>"
                                        "Order Number: %8.8d  Number of Lines:          W_tax:
D_tax:<BR> <BR>"
                                        " Supp_W  Item_Id  Item Name              Qty  Stock  B/G  Price
Amount<BR>"
                                        , pNewOrderData->o_id);

                        i = 0;
                }

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

                if ( bValid )
                        c += sprintf(szForm+c, "Execution Status: Transaction commited.          Total:
$%8.2f  ",
                                        pNewOrderData->total_amount);
                else
```

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

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

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

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

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

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

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

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

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

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

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

```
                                        );

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

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

}

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

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

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

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

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

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

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

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

                strcpy(szForm+c,
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

	PDELIVERY_DATA		pDelivery;

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

	pDelivery->w_id = Term.pClientData[iTermId].w_id;

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

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

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

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

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

	PSTOCK_LEVEL_DATA	pStockLevel;

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

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

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

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

	pStockLevel = Term.pClientData[iTermId].pTxn->BuffAddr_StockLevel();
```

```
	MakeStockLevelForm(iTermId, pStockLevel, OUTPUT_FORM, szBuffer);
}

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

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

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

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

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

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

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

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

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

	pNewOrderData->o_ol_cnt = items;
}

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

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

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

	GetKeyValue(&ptr, "CID*", szTmp, sizeof(szTmp), ERR_PAYMENT_MISSING_CID_KEY);
	if ( szTmp[0] == 0 )
	{
		bCustIdBlank = TRUE;
		pPaymentData->c_id = 0;
	}
	else
```

```
		{		// parse customer id and verify that last name was NOT entered
				bCustIdBlank = FALSE;
				if ( !IsNumeric(szTmp) )
						throw new CWEBCLNT_ERR( ERR_PAYMENT_CUSTOMER_INVALID );
				pPaymentData->c_id = atoi(szTmp);
		}

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

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

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

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

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

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

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

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

}

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

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

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

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

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

        if ( *ptr == 0 )
                return FALSE;

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

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

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

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

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
```

APPENDIX A - APPLICATION SOURCE CODE

```
 *                                                  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);
        }
```

APPENDIX A - APPLICATION SOURCE CODE

```
~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 ADDRESS_LEN             20
#define STATE_LEN               2
#define ZIP_LEN                 9
#define S_DIST_LEN              24
#define S_DATA_LEN              50
#define D_NAME_LEN              10
#define FIRST_NAME_LEN          16
#define MIDDLE_NAME_LEN         2
#define PHONE_LEN               16
#define DATETIME_LEN            30
#define CREDIT_LEN              2
#define C_DATA_LEN              250
#define H_DATA_LEN              24
#define DIST_INFO_LEN           24
#define MAX_OL_NEW_ORDER_ITEMS  15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN              25
#define OL_DIST_INFO_LEN                        24

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

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

APPENDIX A - APPLICATION SOURCE CODE

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

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

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

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

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

        // output params
        EXEC_STATUS                                     exec_status_code;
        TIMESTAMP_STRUCT        h_date;
        char                                    w_street_1[ADDRESS_LEN+1];
        char                                    w_street_2[ADDRESS_LEN+1];
        char                                    w_city[ADDRESS_LEN+1];
        char                                    w_state[STATE_LEN+1];
        char                                    w_zip[ZIP_LEN+1];
        char                                    d_street_1[ADDRESS_LEN+1];
        char                                    d_street_2[ADDRESS_LEN+1];
        char                                    d_city[ADDRESS_LEN+1];
        char                                    d_state[STATE_LEN+1];
        char                            d_zip[ZIP_LEN+1];
        char                                    c_first[FIRST_NAME_LEN+1];
        char                                    c_middle[MIDDLE_NAME_LEN + 1];
        char                                    c_street_1[ADDRESS_LEN+1];
        char                                    c_street_2[ADDRESS_LEN+1];
        char                                    c_city[ADDRESS_LEN+1];
        char                                    c_state[STATE_LEN+1];
        char                                    c_zip[ZIP_LEN+1];
        char                                    c_phone[PHONE_LEN+1];
        TIMESTAMP_STRUCT        c_since;
        char                                    c_credit[CREDIT_LEN+1];
        double                                  c_credit_lim;
        double                                  c_discount;
        double                                  c_balance;
        char                                    c_data[200+1];
} PAYMENT_DATA, *PPAYMENT_DATA;

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

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

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


typedef struct
{       // input params
```

```
        short           w_id;
        short           o_carrier_id;

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

//This structure is used for posting delivery transactions and for writing them to the delivery server.
typedef struct _DELIVERY_TRANSACTION
{
        SYSTEMTIME      queue;                          //time delivery transaction queued
        short           w_id;                           //delivery warehouse
        short           o_carrier_id;  //carrier id
} DELIVERY_TRANSACTION;

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

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

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
```

APPENDIX A - APPLICATION SOURCE CODE

```
#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_SUCH_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
{
        LOGINREC    *login;
        const BYTE    *pData;

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

        m_MaxRetries = 10;              // how many retries on deadlock

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

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

        // register error and message handler functions
        if (dbprocerrhandle(login, err_handler) == NULL)
                ThrowError(CDBLIBERR::eDbProcHandler);

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


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

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

        // set time to wait for statement execution
```

```
        if (dbsettime(180) == FAIL)
                ThrowError(CDBLIBERR::eDbSet);

        m_dbproc = dbopen(login, szServer);

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

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

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

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

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

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

        DiscardNextResults(2);

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

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

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

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

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


        DiscardNextRows(0);
        DiscardNextResults(0);
}


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

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

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

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

}

void CTPCC_DBLIB::SetSqlError( int /*DBINT*/ msgno, int msgstate, int severity, LPCSTR msgtext )
{
        if (m_SqlErr == NULL)
                m_SqlErr = new CSQLERR();

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

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

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

        // check for SQL Server error first;  if yes, throw it and ignore any DBLib error.
        if (m_SqlErr != NULL)
        {
```

```
                CSQLERR         *pSqlErr;
                pSqlErr = m_SqlErr;
                m_SqlErr = NULL;        // clear our pointer to instance; catch handler will delete
                throw pSqlErr;
        }

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

        throw pDbLibErr;
}

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

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

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

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

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

                DiscardNextRows(-1);
                iResultsRead++;
        }

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

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

        ResetError();

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

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

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

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

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

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

                        DiscardNextRows(0);
                        DiscardNextResults(0);

                        m_txn.StockLevel.exec_status_code = eOK;
                        return;
                }
                catch (CSQLERR *e)
                {
                        if ((e->m_msgno != 1205) || (++iTryCount > iMaxRetries))
                                throw;

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


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

        int             iTryCount = 0;
        const BYTE    *pData;

        ResetError();

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

                        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *) &m_txn.NewOrder.w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.NewOrder.d_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.NewOrder.c_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.o_ol_cnt);

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

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

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


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

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

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

                                if(pData=dbdata(m_dbproc, 1))
                                        UtilStrCpy(m_txn.NewOrder.OL[i].ol_i_name, pData,
dbdatlen(m_dbproc, 1));
                                if(pData=dbdata(m_dbproc, 2))
                                        m_txn.NewOrder.OL[i].ol_stock = (*(DBSMALLINT *) pData);
                                if(pData=dbdata(m_dbproc, 3))
                                        UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_generic, pData,
dbdatlen(m_dbproc, 3));
                                if(pData=dbdata(m_dbproc, 4))
                                        dbconvert(m_dbproc, SQLNUMERIC, 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);
                        }
```

APPENDIX A - APPLICATION SOURCE CODE

```
                                        // 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()
{
        DBDATETIME      datetime;
        DBDATEREC       daterec;

        int                             iTryCount = 0;
        const BYTE    *pData;

        ResetError();

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

                        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *) &m_txn.Payment.w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *) &m_txn.Payment.c_w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLFLT8, -1, -1, (BYTE *)
&m_txn.Payment.h_amount);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.Payment.d_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.Payment.c_d_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.Payment.c_id);

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

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

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

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

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

if (pData=dbdata(m_dbproc, 1))
        m_txn.Payment.c_id = *((DBINT *) pData);
if (pData=dbdata(m_dbproc, 2))
        UtilStrCpy(m_txn.Payment.c_last, pData, dbdatlen(m_dbproc, 2));
if (pData=dbdata(m_dbproc, 3))
{
        datetime = *((DBDATETIME *) pData);
        dbdatecrack(m_dbproc, &daterec, &datetime);
        m_txn.Payment.h_date.year   = daterec.year;
        m_txn.Payment.h_date.month  = daterec.month;
        m_txn.Payment.h_date.day    = daterec.day;
        m_txn.Payment.h_date.hour   = daterec.hour;
        m_txn.Payment.h_date.minute = daterec.minute;
        m_txn.Payment.h_date.second = daterec.second;
}
if (pData=dbdata(m_dbproc, 4))
        UtilStrCpy(m_txn.Payment.w_street_1, pData, dbdatlen(m_dbproc, 4));
if (pData=dbdata(m_dbproc, 5))
        UtilStrCpy(m_txn.Payment.w_street_2, pData, dbdatlen(m_dbproc, 5));
if (pData=dbdata(m_dbproc, 6))
        UtilStrCpy(m_txn.Payment.w_city, pData, dbdatlen(m_dbproc, 6));
if (pData=dbdata(m_dbproc, 7))
        UtilStrCpy(m_txn.Payment.w_state, pData, dbdatlen(m_dbproc, 7));
if (pData=dbdata(m_dbproc, 8))
        UtilStrCpy(m_txn.Payment.w_zip, pData, dbdatlen(m_dbproc, 8));
if (pData=dbdata(m_dbproc, 9))
        UtilStrCpy(m_txn.Payment.d_street_1, pData, dbdatlen(m_dbproc, 9));
if (pData=dbdata(m_dbproc, 10))
        UtilStrCpy(m_txn.Payment.d_street_2, pData, dbdatlen(m_dbproc, 10));
if (pData=dbdata(m_dbproc, 11))
        UtilStrCpy(m_txn.Payment.d_city, pData, dbdatlen(m_dbproc, 11));
if (pData=dbdata(m_dbproc, 12) )
        UtilStrCpy(m_txn.Payment.d_state, pData, dbdatlen(m_dbproc, 12));
if (pData=dbdata(m_dbproc, 13))
        UtilStrCpy(m_txn.Payment.d_zip, pData, dbdatlen(m_dbproc, 13));
if (pData=dbdata(m_dbproc, 14))
        UtilStrCpy(m_txn.Payment.c_first, pData, dbdatlen(m_dbproc, 14));
if (pData=dbdata(m_dbproc, 15))
        UtilStrCpy(m_txn.Payment.c_middle, pData, dbdatlen(m_dbproc, 15));
if (pData=dbdata(m_dbproc, 16))
        UtilStrCpy(m_txn.Payment.c_street_1, pData, dbdatlen(m_dbproc, 16));
if (pData=dbdata(m_dbproc, 17))
        UtilStrCpy(m_txn.Payment.c_street_2, pData, dbdatlen(m_dbproc, 17));
if (pData=dbdata(m_dbproc, 18))
        UtilStrCpy(m_txn.Payment.c_city, pData, dbdatlen(m_dbproc, 18));
if (pData=dbdata(m_dbproc, 19))
        UtilStrCpy(m_txn.Payment.c_state, pData, dbdatlen(m_dbproc, 19));
if (pData=dbdata(m_dbproc, 20))
        UtilStrCpy(m_txn.Payment.c_zip, pData, dbdatlen(m_dbproc, 20));
if (pData=dbdata(m_dbproc, 21))
        UtilStrCpy(m_txn.Payment.c_phone, pData, dbdatlen(m_dbproc, 21));
if (pData=dbdata(m_dbproc, 22))
{
        datetime = *((DBDATETIME *) pData);
        dbdatecrack(m_dbproc, &daterec, &datetime);
        m_txn.Payment.c_since.year   = daterec.year;
        m_txn.Payment.c_since.month  = daterec.month;
        m_txn.Payment.c_since.day    = daterec.day;
        m_txn.Payment.c_since.hour   = daterec.hour;
        m_txn.Payment.c_since.minute = daterec.minute;
        m_txn.Payment.c_since.second = daterec.second;
}
if(pData=dbdata(m_dbproc, 23))
        UtilStrCpy(m_txn.Payment.c_credit, pData, dbdatlen(m_dbproc, 23));
if(pData=dbdata(m_dbproc, 24))
        dbconvert(m_dbproc, SQLNUMERIC, 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;
    DBDATETIME          datetime;
```

```
	DBDATEREC		daterec;

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

	ResetError();

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

			dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *) &m_txn.OrderStatus.w_id);
			dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.OrderStatus.d_id);
			dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.OrderStatus.c_id);

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


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

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

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

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

				if(pData=dbdata(m_dbproc, 1))
					m_txn.OrderStatus.OL[i].ol_supply_w_id = (*(DBSMALLINT
*) pData);
				if(pData=dbdata(m_dbproc, 2))
					m_txn.OrderStatus.OL[i].ol_i_id = (*(DBINT *) pData);
				if(pData=dbdata(m_dbproc, 3))
					m_txn.OrderStatus.OL[i].ol_quantity = (*(DBSMALLINT *)
pData);
				if(pData=dbdata(m_dbproc, 4))
					dbconvert(m_dbproc, SQLNUMERIC, 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(CDBLIBER R::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))
```

APPENDIX A - APPLICATION SOURCE CODE

```
                                        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

APPENDIX A - APPLICATION SOURCE CODE

```
/*      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 definations of structures specific to
TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_com.h"

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

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

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

        m_bSinglePool = bSinglePool;

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

        m_pTxn = (COM_DATA*)CoTaskMemAlloc(sizeof(COM_DATA));
        if (!m_pTxn)
                throw new CCOMERR( E_FAIL );

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

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

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

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

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

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

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

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

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

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

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

CTPCC_COM::~CTPCC_COM()
{
        if (m_pTxn)
                CoTaskMemFree(m_pTxn);

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

void CTPCC_COM::NewOrder()
{
        int iSize = sizeof(COM_DATA);

        HRESULT hr = m_pNewOrder->NewOrder(&iSize, (unsigned char**)&m_pTxn);
        if (FAILED(hr))
                throw new CCOMERR( hr );

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

void CTPCC_COM::Payment()
{
        int iSize = sizeof(COM_DATA);

        HRESULT hr = m_pPayment->Payment(&iSize, (unsigned char**)&m_pTxn);
        if (FAILED(hr))
                throw new CCOMERR( hr );

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

void CTPCC_COM::StockLevel()
{
        int iSize = sizeof(COM_DATA);

        HRESULT hr = m_pStockLevel->StockLevel(&iSize, (unsigned char**)&m_pTxn);
        if (FAILED(hr))
                throw new CCOMERR( hr );

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

void CTPCC_COM::OrderStatus()
{
        int iSize = sizeof(COM_DATA);

        HRESULT hr = m_pOrderStatus->OrderStatus(&iSize, (unsigned char**)&m_pTxn);
        if (FAILED(hr))
                throw new CCOMERR( hr );

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

tpcc_com_all\src\methods.h

APPENDIX A - APPLICATION SOURCE CODE

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


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

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

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

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

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

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


static void WriteMessageToEventLog(LPTSTR lpszMsg);


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

        CTPCC_Common();
        ~CTPCC_Common();

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

        HRESULT __stdcall CallSetComplete();

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

// IObjectConstruct
        STDMETHODIMP Construct(IDispatch * pUnk);

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

        struct COM_DATA
        {
```

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

};

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

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

};


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

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

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


        //////////////////////////////////////////////////////////////////

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

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

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


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

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

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


//////////////////////////////////////////////////////////////////
// CStockLevel
class CStockLevel :
        public CTPCC_Common,
        public CComCoClass<CStockLevel, &CLSID_StockLevel>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_STOCKLEVEL)

BEGIN_COM_MAP(CStockLevel)
```

```
		COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
		COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

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

tpcc_com_all\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
```

tpcc_com_all\src\tpcc_com_all.cpp

```
/*		FILE:			TPCC_COM_ALL.CPP
 *					Microsoft TPC-C Kit Ver. 4.20.000
 *					Copyright Microsoft, 1999
 *	All Rights Reserved
 *
 *					Version 4.10.000 audited by Richard Gimarc, Performance Metrics,
3/17/99
 *
 *		PURPOSE:	Implementation for TPC-C Tuxedo class.
 *		Contact:	Charles Levine (clevine@microsoft.com)
 *
 * Change history:
 *			4.20.000 - updated rev number to match kit
 */

#define STRICT
#define _WIN32_WINNT 0x0400
#define _ATL_APARTMENT_THREADED

#include <stdio.h>
#include <atlbase.h>
//You may derive a class from CComModule and use it if you want to override
//something, but do not change the name of _Module
extern CComModule _Module;

#include <atlcom.h>
#include <initguid.h>
#include <transact.h>
#include <atlimpl.cpp>
#include <comsvcs.h>

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

#include "tpcc_com_ps.h"
#include "..\..\common\src\trans.h"                                    //tpckit transaction header
contains definations of structures specific to TPC-C
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\error.h"
#include "..\..\common\src\ReadRegistry.h"
#include "..\..\db_dblib_dll\src\tpcc_dblib.h"                  // DBLIB implementation of TPC-C txns
#include "..\..\db_odbc_dll\src\tpcc_odbc.h"                    // ODBC implementation of TPC-C txns


#include "resource.h"
#include "tpcc_com_all.h"
#include "tpcc_com_all_i.c"
#include "Methods.h"
#include "..\..\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "..\..\common\src\ReadRegistry.cpp"


CComModule _Module;

BEGIN_OBJECT_MAP(ObjectMap)
		OBJECT_ENTRY(CLSID_TPCC, CTPCC)
		OBJECT_ENTRY(CLSID_NewOrder, CNewOrder)
		OBJECT_ENTRY(CLSID_OrderStatus, COrderStatus)
		OBJECT_ENTRY(CLSID_Payment, CPayment)
		OBJECT_ENTRY(CLSID_StockLevel, CStockLevel)
END_OBJECT_MAP()
```

APPENDIX A - APPLICATION SOURCE CODE

```
// configuration settings from registry
TPCCREGISTRYDATA	Reg;
char				szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

static  HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB	*pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC		*pCTPCC_ODBC_new;


////////////////////////////////////////////////////////////////////
// DLL Entry Point

extern "C"
BOOL WINAPI DllMain(HINSTANCE hInstance, DWORD dwReason, LPVOID /*lpReserved*/)
{
	char szDllName[128];

	try
	{
		if (dwReason == DLL_PROCESS_ATTACH)
		{
			_Module.Init(ObjectMap, hInstance);
			DisableThreadLibraryCalls(hInstance);

			DWORD dwSize = MAX_COMPUTERNAME_LENGTH+1;
			GetComputerName(szMyComputerName, &dwSize);
			szMyComputerName[dwSize] = 0;

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

			if (Reg.eDB_Protocol == DBLIB)
			{
				strcpy( szDllName, Reg.szPath );
				strcat( szDllName, "tpcc_dblib.dll");
				hLibInstanceDb = LoadLibrary( szDllName );
				if (hLibInstanceDb == NULL)
					throw new CCOMPONENT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );

				// get function pointer to wrapper for class constructor
				pCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_new");
				if (pCTPCC_DBLIB_new == NULL)
					throw new CCOMPONENT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
			}
			else if (Reg.eDB_Protocol == ODBC)
			{
				strcpy( szDllName, Reg.szPath );
				strcat( szDllName, "tpcc_odbc.dll");
				hLibInstanceDb = LoadLibrary( szDllName );
				if (hLibInstanceDb == NULL)
					throw new CCOMPONENT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );

				// get function pointer to wrapper for class constructor
				pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
				if (pCTPCC_ODBC_new == NULL)
					throw new CCOMPONENT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
			}
			else
				throw new CCOMPONENT_ERR( ERR_UNKNOWN_DB_PROTOCOL );
		}
		else if (dwReason == DLL_PROCESS_DETACH)
			_Module.Term();

	}
	catch (CBaseErr *e)
	{
		WriteMessageToEventLog(e->ErrorText());
		delete e;
		return FALSE;
	}
	catch (...)
	{
		WriteMessageToEventLog(TEXT("Unhandled exception in object DllMain"));
		return FALSE;
	}

	return TRUE; // OK
}

////////////////////////////////////////////////////////////////////
// Used to determine whether the DLL can be unloaded by OLE

STDAPI DllCanUnloadNow(void)
{
	return (_Module.GetLockCount()==0) ? S_OK : S_FALSE;
}

////////////////////////////////////////////////////////////////////
// Returns a class factory to create an object of the requested type

STDAPI DllGetClassObject(REFCLSID rclsid, REFIID riid, LPVOID* ppv)
{
	return _Module.GetClassObject(rclsid, riid, ppv);
}

////////////////////////////////////////////////////////////////////
// DllRegisterServer - Adds entries to the system registry

STDAPI DllRegisterServer(void)
{
	// registers object, typelib and all interfaces in typelib
	return _Module.RegisterServer(TRUE);
```

```
}

////////////////////////////////////////////////////////////////////
// DllUnregisterServer - Removes entries from the system registry

STDAPI DllUnregisterServer(void)
{
        _Module.UnregisterServer();
        return S_OK;
}


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

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

   _stprintf(szMsg, TEXT("Error in COM+ TPC-C Component: "));
   lpszStrings[0] = szMsg;
   lpszStrings[1] = lpszMsg;

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

     (VOID) DeregisterEventSource(hEventSource);
   }
}

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

/* FUNCTION: CCOMPONENT_ERR::ErrorText
 *
 */
char* CCOMPONENT_ERR::ErrorText(void)
{
        static SERRORMSG errorMsgs[] =
        {
                { ERR_MISSING_REGISTRY_ENTRIES,     "Required entries missing from registry."
                        },
                { ERR_LOADDLL_FAILED,                       "Load of DLL failed.  DLL="
                                                                        },
                { ERR_GETPROCADDR_FAILED,           "Could not map proc in DLL.  GetProcAddr
error.  DLL=" },
                { ERR_UNKNOWN_DB_PROTOCOL,          "Unknown database protocol specified in
registry."      },
                { 0,                                                                    ""
                                                                        }
        };

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

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

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


CTPCC_Common::CTPCC_Common()
{
        m_pTxn = NULL;
        m_bCanBePooled = TRUE;
}

CTPCC_Common::~CTPCC_Common()
{
        if (m_pTxn)
                delete m_pTxn;
```

```
}

HRESULT CTPCC_Common::CallSetComplete()
{
	IObjectContext* pObjectContext = NULL;

	// get our object context
	HRESULT hr = CoGetObjectContext( IID_IObjectContext, (void **)&pObjectContext );
	pObjectContext->SetComplete();
	ReleaseInterface(pObjectContext);
	return hr;
}

//
//	called by the ctor activator
//
STDMETHODIMP CTPCC_Common::Construct(IDispatch * pUnk)
{
	// Code to access construction string, if needed later...
	//	if (!pUnk)
	//		return E_UNEXPECTED;
	//	IObjectConstructString * pString = NULL;
	//	HRESULT hr = pUnk->QueryInterface(IID_IObjectConstructString, (void **)&pString);
	//	pString->Release();

	try
	{
		if (Reg.eDB_Protocol == ODBC)
			m_pTxn = pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );
		else if (Reg.eDB_Protocol == DBLIB)
			m_pTxn = pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );
	}
	catch (CBaseErr *e)
	{
		WriteMessageToEventLog(e->ErrorText());
		delete e;
		return E_FAIL;
	}
	catch (...)
	{
		WriteMessageToEventLog(TEXT("Unhandled exception in object ::Construct"));
		return E_FAIL;
	}

	return S_OK;
}

HRESULT CTPCC_Common::NewOrder(int* iSize, UCHAR **txn)
{
	PNEW_ORDER_DATA	pNewOrder;
	COM_DATA		*pData;

	try
	{
		pData = (COM_DATA*)*txn;
		pNewOrder = m_pTxn->BuffAddr_NewOrder();

		memcpy(pNewOrder, &pData->u.NewOrder, sizeof(NEW_ORDER_DATA));
		m_pTxn->NewOrder();
		memcpy( &pData->u.NewOrder, pNewOrder, sizeof(NEW_ORDER_DATA));

		pData->retval = ERR_SUCCESS;
		pData->error = 0;
		return S_OK;
	}
	catch (CBaseErr *e)
	{
		// check for lost database connection; if yes, component is toast
		if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
			((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
			m_bCanBePooled = FALSE;

		pData->retval = e->ErrorType();
		pData->error = e->ErrorNum();
		delete e;
		return E_FAIL;
	}
	catch (...)
	{
		WriteMessageToEventLog(TEXT("Unhandled exception."));
		pData->retval = ERR_TYPE_LOGIC;
		pData->error = 0;
		m_bCanBePooled = FALSE;
		return E_FAIL;
	}
}

HRESULT CTPCC_Common::Payment(int* iSize, UCHAR** txn)
{
	PPAYMENT_DATA	pPayment;
	COM_DATA		*pData;

	try
	{
		pData = (COM_DATA*)*txn;
		pPayment = m_pTxn->BuffAddr_Payment();

		memcpy(pPayment, &pData->u.Payment, sizeof(PAYMENT_DATA) );
		m_pTxn->Payment();
		memcpy( &pData->u.Payment, pPayment, sizeof(PAYMENT_DATA) );

		pData->retval = ERR_SUCCESS;
		pData->error = 0;
		return S_OK;
	}
	catch (CBaseErr *e)
	{
		// check for lost database connection; if yes, component is toast
```

```
                if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
                        ((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                        m_bCanBePooled = FALSE;

                pData->retval = e->ErrorType();
                pData->error = e->ErrorNum();
                delete e;
                return E_FAIL;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception."));
                pData->retval = ERR_TYPE_LOGIC;
                pData->error = 0;
                m_bCanBePooled = FALSE;
                return E_FAIL;
        }
}

HRESULT CTPCC_Common::StockLevel(int* iSize, UCHAR** txn)
{
        PSTOCK_LEVEL_DATA     pStockLevel;
        COM_DATA                        *pData;

        try
        {
                pData = (COM_DATA*)*txn;
                pStockLevel = m_pTxn->BuffAddr_StockLevel();

                memcpy(pStockLevel, &pData->u.StockLevel, sizeof(STOCK_LEVEL_DATA) );
                m_pTxn->StockLevel();
                memcpy( &pData->u.StockLevel, pStockLevel, sizeof(STOCK_LEVEL_DATA) );

                pData->retval = ERR_SUCCESS;
                pData->error = 0;
                return S_OK;
        }
        catch (CBaseErr *e)
        {
                // check for lost database connection; if yes, component is toast
                if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
                        ((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                        m_bCanBePooled = FALSE;

                pData->retval = e->ErrorType();
                pData->error = e->ErrorNum();
                delete e;
                return E_FAIL;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception."));
                pData->retval = ERR_TYPE_LOGIC;
                pData->error = 0;
                m_bCanBePooled = FALSE;
                return E_FAIL;
        }
}

HRESULT CTPCC_Common::OrderStatus(int* iSize, UCHAR** txn)
{
        PORDER_STATUS_DATA              pOrderStatus;
        COM_DATA                        *pData;

        try
        {
                pData = (COM_DATA*)*txn;
                pOrderStatus = m_pTxn->BuffAddr_OrderStatus();

                memcpy(pOrderStatus, &pData->u.OrderStatus, sizeof(ORDER_STATUS_DATA) );
                m_pTxn->OrderStatus();
                memcpy( &pData->u.OrderStatus, pOrderStatus, sizeof(ORDER_STATUS_DATA) );

                pData->retval = ERR_SUCCESS;
                pData->error = 0;
                return S_OK;
        }
        catch (CBaseErr *e)
        {
                // check for lost database connection; if yes, component is toast
                if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
                        ((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                        m_bCanBePooled = FALSE;

                pData->retval = e->ErrorType();
                pData->error = e->ErrorNum();
                delete e;
                return E_FAIL;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception."));
                pData->retval = ERR_TYPE_LOGIC;
                pData->error = 0;
                m_bCanBePooled = FALSE;
                return E_FAIL;
        }
}
```

tpcc_com_all\src\tpcc_com_all.def

```
; tpcc_com_all.def : Declares the module parameters.

LIBRARY     "tpcc_com_all.dll"
```

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

tpcc_com_all\src\tpcc_com_all.h

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

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


 /* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jan 24 20:00:20 2000
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
        DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


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

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

#ifndef __tpcc_com_all_h__
#define __tpcc_com_all_h__

/* Forward Declarations */

#ifndef __TPCC_FWD_DEFINED__
#define __TPCC_FWD_DEFINED__

#ifdef __cplusplus
typedef class TPCC TPCC;
#else
typedef struct TPCC TPCC;
#endif /* __cplusplus */

#endif          /* __TPCC_FWD_DEFINED__ */


#ifndef __NewOrder_FWD_DEFINED__
#define __NewOrder_FWD_DEFINED__

#ifdef __cplusplus
typedef class NewOrder NewOrder;
#else
typedef struct NewOrder NewOrder;
#endif /* __cplusplus */

#endif          /* __NewOrder_FWD_DEFINED__ */


#ifndef __OrderStatus_FWD_DEFINED__
#define __OrderStatus_FWD_DEFINED__

#ifdef __cplusplus
typedef class OrderStatus OrderStatus;
#else
typedef struct OrderStatus OrderStatus;
#endif /* __cplusplus */

#endif          /* __OrderStatus_FWD_DEFINED__ */


#ifndef __Payment_FWD_DEFINED__
#define __Payment_FWD_DEFINED__

#ifdef __cplusplus
typedef class Payment Payment;
#else
typedef struct Payment Payment;
#endif /* __cplusplus */

#endif          /* __Payment_FWD_DEFINED__ */


#ifndef __StockLevel_FWD_DEFINED__
#define __StockLevel_FWD_DEFINED__

#ifdef __cplusplus
typedef class StockLevel StockLevel;
#else
typedef struct StockLevel StockLevel;
#endif /* __cplusplus */

#endif          /* __StockLevel_FWD_DEFINED__ */


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

#ifdef __cplusplus
extern "C"{
```

```
#endif

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

/* interface __MIDL_itf_tpcc_com_all_0000 */
/* [local] */

extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_s_ifspec;

#ifndef __TPCCLib_LIBRARY_DEFINED__
#define __TPCCLib_LIBRARY_DEFINED__

/* library TPCCLib */
/* [helpstring][version][uuid] */
EXTERN_C const IID LIBID_TPCCLib;

EXTERN_C const CLSID CLSID_TPCC;

#ifdef __cplusplus

class DECLSPEC_UUID("122A3128-2520-11D3-BA71-00C04FBFE08B")
TPCC;
#endif

EXTERN_C const CLSID CLSID_NewOrder;

#ifdef __cplusplus

class DECLSPEC_UUID("975BAABF-84A7-11D2-BA47-00C04FBFE08B")
NewOrder;
#endif

EXTERN_C const CLSID CLSID_OrderStatus;

#ifdef __cplusplus

class DECLSPEC_UUID("266836AD-A50D-11D2-BA4E-00C04FBFE08B")
OrderStatus;
#endif

EXTERN_C const CLSID CLSID_Payment;

#ifdef __cplusplus

class DECLSPEC_UUID("CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B")
Payment;
#endif

EXTERN_C const CLSID CLSID_StockLevel;

#ifdef __cplusplus

class DECLSPEC_UUID("2668369E-A50D-11D2-BA4E-00C04FBFE08B")
StockLevel;
#endif
#endif /* __TPCCLib_LIBRARY_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

pcc_com_all\src\tpcc_com_all.idl

```
/*          FILE:                   TPCC.IDL
 *                                          Microsoft TPC-C Kit Ver. 4.20.000
 *                                          Copyright Microsoft, 1999
 *      All Rights Reserved
 *
 *                                          not yet audited
 *
 *          PURPOSE:    IDL source for TPCC.dll.  This file is processed by the MIDL tool to
 *                                          produce the type library (TPCC.tlb) and marshalling code.
 *
 *  Change history:
 *                      4.20.000 - first version
 */

interface TPCC;
interface NewOrder;
interface OrderStatus;
interface Payment;
interface StockLevel;


import "oaidl.idl";
import "ocidl.idl";
import "..\tpcc_com_ps\src\tpcc_com_ps.idl";

[
        uuid(122A3117-2520-11D3-BA71-00C04FBFE08B),
        version(1.0),
        helpstring("TPC-C 1.0 Type Library")
]
library TPCCLib
{
        importlib("stdole32.tlb");
        importlib("stdole2.tlb");
```

```
	[
		uuid(122A3128-2520-11D3-BA71-00C04FBFE08B),
		helpstring("All Txns Class")
	]
	coclass TPCC
	{
		[default] interface ITPCC;
	};


	[
		uuid(975BAABF-84A7-11D2-BA47-00C04FBFE08B),
		helpstring("NewOrder Class")
	]
	coclass NewOrder
	{
		[default] interface ITPCC;
	};


	[
		uuid(266836AD-A50D-11D2-BA4E-00C04FBFE08B),
		helpstring("OrderStatus Class")
	]
	coclass OrderStatus
	{
		[default] interface ITPCC;
	};


	[
		uuid(CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B),
		helpstring("Payment Class")
	]
	coclass Payment
	{
		[default] interface ITPCC;
	};
	[
		uuid(2668369E-A50D-11D2-BA4E-00C04FBFE08B),
		helpstring("StockLevel Class")
	]
	coclass StockLevel
	{
		[default] interface ITPCC;
	};
};
```

tpcc_com_all\src\tpcc_com_all.rc

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

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

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

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

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

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

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

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

3 TEXTINCLUDE DISCARDABLE
BEGIN
   "1 TYPELIB ""tpcc_com_all.tlb""\r\n"
   "\0"
END

#endif    // APSTUDIO_INVOKED


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

VS_VERSION_INFO VERSIONINFO
```

```
 FILEVERSION 1,0,0,1
 PRODUCTVERSION 1,0,0,1
 FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
 FILEFLAGS 0x1L
#else
 FILEFLAGS 0x0L
#endif
 FILEOS 0x4L
 FILETYPE 0x2L
 FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904B0"
        BEGIN
            VALUE "CompanyName", "\0"
            VALUE "FileDescription", "tpcc_com_all Module\0"
            VALUE "FileVersion", "1, 0, 0, 1\0"
            VALUE "InternalName", "TPCCNEWORDER\0"
            VALUE "LegalCopyright", "Copyright 1997\0"
            VALUE "OriginalFilename", "tpcc_com_all.DLL\0"
            VALUE "ProductName", "tpcc_com_all Module\0"
            VALUE "ProductVersion", "1, 0, 0, 1\0"
            VALUE "OLESelfRegister", "\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END

#endif    // !_MAC
/////////////////////////////////////////////////////////////////////////////
//
// REGISTRY
//

IDR_TPCC              REGISTRY DISCARDABLE    "tpcc_com_all.rgs"
IDR_NEWORDER          REGISTRY DISCARDABLE    "tpcc_com_no.rgs"
IDR_ORDERSTATUS       REGISTRY DISCARDABLE    "tpcc_com_os.rgs"
IDR_PAYMENT           REGISTRY DISCARDABLE    "tpcc_com_pay.rgs"
IDR_STOCKLEVEL        REGISTRY DISCARDABLE    "tpcc_com_sl.rgs"

/////////////////////////////////////////////////////////////////////////////
//
// String Table
//

STRINGTABLE DISCARDABLE
BEGIN
    IDS_PROJNAME          "tpcc_com_all"
END

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



#ifndef APSTUDIO_INVOKED
/////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//
1 TYPELIB "tpcc_com_all.tlb"

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

tpcc_common_all\src\tpcc_com_all.rgs

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

tpcc_common_all\src\tpcc_com_all_i.c

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

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

APPENDIX A - APPLICATION SOURCE CODE

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


 /* File created by MIDL compiler version 5.02.0235 */
/* at Fri Aug 13 18:56:24 1999
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

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

#ifdef __cplusplus
extern "C"{
#endif


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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID, LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID, CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID, CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus,0x266836AD,0xA50D ,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID, CLSID_Payment,0xCD02F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID, CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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


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

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

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


 /* File created by MIDL compiler version 5.02.0235 */
/* at Fri Aug 13 18:56:25 1999
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), 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)
```

APPENDIX A - APPLICATION SOURCE CODE

```
#ifdef __cplusplus
extern "C"{
#endif


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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID, LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID, CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID, CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID, CLSID_Payment,0xCD02F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID, CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

tpcc_common_all\src\tpcc_com_no.rgs

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

tpcc_common_all\src\tpcc_com_os.rgs

```
HKCR
```

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

tpcc_common_all\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 5.02.0235 */
/* at Fri Aug 13 18:56:17 1999
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


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

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

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

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

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

/* Forward Declarations */

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


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

#ifdef __cplusplus
extern "C"{
#endif

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

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

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


EXTERN_C const IID IID_ITPCC;

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

    MIDL_INTERFACE("FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
    ITPCC : public IUnknown
    {
    public:
        virtual HRESULT __stdcall NewOrder(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn) = 0;

        virtual HRESULT __stdcall Payment(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn) = 0;

        virtual HRESULT __stdcall Delivery(
```

```
        /* [in] */ int __RPC_FAR *iSize,
        /* [size_is][size_is][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn) = 0;

    virtual HRESULT __stdcall StockLevel(
        /* [out][in] */ int __RPC_FAR *iSize,
        /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn) = 0;

    virtual HRESULT __stdcall OrderStatus(
        /* [out][in] */ int __RPC_FAR *iSize,
        /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn) = 0;

    virtual HRESULT __stdcall CallSetComplete( void) = 0;

  };

#else           /* C style interface */

  typedef struct ITPCCVtbl
  {
    BEGIN_INTERFACE

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

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

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

    HRESULT ( __stdcall __RPC_FAR *NewOrder )(
        ITPCC __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR *iSize,
        /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

    HRESULT ( __stdcall __RPC_FAR *Payment )(
        ITPCC __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR *iSize,
        /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

    HRESULT ( __stdcall __RPC_FAR *Delivery )(
        ITPCC __RPC_FAR * This,
        /* [in] */ int __RPC_FAR *iSize,
        /* [size_is][size_is][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

    HRESULT ( __stdcall __RPC_FAR *StockLevel )(
        ITPCC __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR *iSize,
        /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

    HRESULT ( __stdcall __RPC_FAR *OrderStatus )(
        ITPCC __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR *iSize,
        /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

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

    END_INTERFACE
  } ITPCCVtbl;

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


#ifdef COBJMACROS


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

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

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


#define ITPCC_NewOrder(This,iSize,txn)                      \
    (This)->lpVtbl -> NewOrder(This,iSize,txn)

#define ITPCC_Payment(This,iSize,txn)       \
    (This)->lpVtbl -> Payment(This,iSize,txn)

#define ITPCC_Delivery(This,iSize,txn)      \
    (This)->lpVtbl -> Delivery(This,iSize,txn)

#define ITPCC_StockLevel(This,iSize,txn)                    \
    (This)->lpVtbl -> StockLevel(This,iSize,txn)

#define ITPCC_OrderStatus(This,iSize,txn)                   \
    (This)->lpVtbl -> OrderStatus(This,iSize,txn)

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

#endif /* COBJMACROS */


#endif          /* C style interface */


HRESULT __stdcall ITPCC_NewOrder_Proxy(
    ITPCC __RPC_FAR * This,
```

```
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);


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


HRESULT __stdcall ITPCC_Payment_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);


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


HRESULT __stdcall ITPCC_Delivery_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);


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


HRESULT __stdcall ITPCC_StockLevel_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);


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


HRESULT __stdcall ITPCC_OrderStatus_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);


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


HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC __RPC_FAR * This);


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

#endif          /* __ITPCC_INTERFACE_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

tpcc_common_all\src\tpcc_com_pay.rgs

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

```
                                        }
                                }
                        }
}
```

tpcc_common_all\src\tpcc_com_sl.rgs

```
HKCR
{
        TPCC.StockLevel.1 = s 'StockLevel Class'
        {
                CLSID = s '{2668369E-A50D-11D2-BA4E-00C04FBFE08B}'
        }
        TPCC.StockLevel = s 'StockLevel Class'
        {
                CurVer = s 'TPCC.StockLevel.1'
        }
        NoRemove CLSID
        {
                ForceRemove {2668369E-A50D-11D2-BA4E-00C04FBFE08B} = s 'StockLevel Class'
                {
                        ProgID = s 'TPCC.StockLevel.1'
                        VersionIndependentProgID = s 'TPCC.StockLevel'
                        InprocServer32 = s '%MODULE%'
                        {
                                val ThreadingModel = s 'Both'
                        }
                }
        }
}
```

tpcc_common_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_common_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 5.02.0235 */
/* at Fri Aug 13 18:56:17 1999
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
        DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


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

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

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

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

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

/* Forward Declarations */

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


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

#ifdef __cplusplus
extern"C"{
```

```
#endif

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

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

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


EXTERN_C const IID IID_ITPCC;

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

    MIDL_INTERFACE("FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
    ITPCC : public IUnknown
    {
    public:
        virtual HRESULT __stdcall NewOrder(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn) = 0;

        virtual HRESULT __stdcall Payment(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn) = 0;

        virtual HRESULT __stdcall Delivery(
            /* [in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn) = 0;

        virtual HRESULT __stdcall StockLevel(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn) = 0;

        virtual HRESULT __stdcall OrderStatus(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn) = 0;

        virtual HRESULT __stdcall CallSetComplete( void) = 0;

    };

#else       /* C style interface */

    typedef struct ITPCCVtbl
    {
        BEGIN_INTERFACE

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

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

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

        HRESULT ( __stdcall __RPC_FAR *NewOrder )(
            ITPCC __RPC_FAR * This,
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

        HRESULT ( __stdcall __RPC_FAR *Payment )(
            ITPCC __RPC_FAR * This,
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

        HRESULT ( __stdcall __RPC_FAR *Delivery )(
            ITPCC __RPC_FAR * This,
            /* [in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

        HRESULT ( __stdcall __RPC_FAR *StockLevel )(
            ITPCC __RPC_FAR * This,
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

        HRESULT ( __stdcall __RPC_FAR *OrderStatus )(
            ITPCC __RPC_FAR * This,
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

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

        END_INTERFACE
    } ITPCCVtbl;

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



#ifdef COBJMACROS


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

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

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

```
#define ITPCC_NewOrder(This,iSize,txn)                  \
    (This)->lpVtbl -> NewOrder(This,iSize,txn)

#define ITPCC_Payment(This,iSize,txn)     \
    (This)->lpVtbl -> Payment(This,iSize,txn)

#define ITPCC_Delivery(This,iSize,txn)     \
    (This)->lpVtbl -> Delivery(This,iSize,txn)

#define ITPCC_StockLevel(This,iSize,txn)                  \
    (This)->lpVtbl -> StockLevel(This,iSize,txn)

#define ITPCC_OrderStatus(This,iSize,txn)                  \
    (This)->lpVtbl -> OrderStatus(This,iSize,txn)

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

#endif /* COBJMACROS */


#endif          /* C style interface */



HRESULT __stdcall ITPCC_NewOrder_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);


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


HRESULT __stdcall ITPCC_Payment_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);


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


HRESULT __stdcall ITPCC_Delivery_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);


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


HRESULT __stdcall ITPCC_StockLevel_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);


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


HRESULT __stdcall ITPCC_OrderStatus_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);


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


HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC __RPC_FAR * This);


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



#endif          /* __ITPCC_INTERFACE_DEFINED__ */


/* Additional Prototypes for ALL interfaces */

/* end of Additional Prototypes */
```

```
#ifdef __cplusplus
}
#endif

#endif
```

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

    HRESULT _stdcall NewOrder
                                                                    (
                                                                    [in, out] int* iSize,
                                                                    [in, out, size_is( , *iSize)] char** txn
                                                                    );

    HRESULT _stdcall Payment
                                                                    (
                                                                    [in, out] int* iSize,
                                                                    [in, out, size_is( , *iSize)] char** txn
                                                                    );

    HRESULT _stdcall Delivery
                                                                    (
                                                                    [in] int* iSize,
                                                                    [in, size_is( , *iSize)] char** txn
                                                                    );

    HRESULT _stdcall StockLevel
                                                                    (
                                                                    [in, out] int* iSize,
                                                                    [in, out, size_is( , *iSize)] char** txn
                                                                    );

    HRESULT _stdcall OrderStatus
                                                                    (
                                                                    [in, out] int* iSize,
                                                                    [in, out, size_is( , *iSize)] char** txn
                                                                    );

    HRESULT _stdcall CallSetComplete
                                                                    (
                                                                    );

};          // interface ITPCC
```

tpcc_common_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 5.02.0235 */
/* at Fri Aug 13 18:56:17 1999
 */
/* 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>
```

APPENDIX A - APPLICATION SOURCE CODE

```
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



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


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

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

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


 /* File created by MIDL compiler version 5.02.0235 */
/* at Fri Aug 13 18:56:18 1999
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

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

#ifdef __cplusplus
extern "C"{
#endif


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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__
```

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

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

#endif !_MIDL_USE_GUIDDEF_

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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



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

tpcc_common_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 5.02.0235 */
/* at Fri Aug 13 18:56:17 1999
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

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


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

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


#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE   33
#define PROC_FORMAT_STRING_SIZE   193
#define TRANSMIT_AS_TABLE_SIZE    0
#define WIRE_MARSHAL_TABLE_SIZE   0

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

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


extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;


/* Object interface: IUnknown, ver. 0.0,
   GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */


/* Object interface: ITPCC, ver. 0.0,
   GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,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
};


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,
    0x50200eb, /* MIDL Version 5.2.235 */
    0,
    0,
    0,  /* notify & notify_flag routine table */
    1,  /* Flags */
    0,  /* Reserved3 */
    0,  /* Reserved4 */
    0   /* Reserved5 */
    };

#pragma data_seg(".rdata")

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

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


static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
    {
        0,
        {

                /* Procedure NewOrder */

                                        0x33,               /* FC_AUTO_HANDLE */
                                        0x6c,               /* Old Flags:  object, Oi2 */
/*  2 */        NdrFcLong( 0x0 ),       /* 0 */
/*  6 */        NdrFcShort( 0x3 ),      /* 3 */
#ifndef _ALPHA_
/*  8 */        NdrFcShort( 0x10 ),     /* x86, MIPS, PPC Stack size/offset = 16 */
#else
                                        NdrFcShort( 0x20 ),     /* Alpha Stack size/offset = 32 */
#endif
/* 10 */        NdrFcShort( 0x8 ),      /* 8 */
/* 12 */        NdrFcShort( 0x10 ),     /* 16 */
/* 14 */        0x7,                    /* Oi2 Flags:  srv must size, clt must size, has return, */
                                        0x3,                /* 3 */

                /* Parameter iSize */

/* 16 */        NdrFcShort( 0x158 ),    /* Flags:  in, out, base type, simple ref, */
#ifndef _ALPHA_
/* 18 */        NdrFcShort( 0x4 ),      /* x86, MIPS, PPC Stack size/offset = 4 */
```

```
#else
                                NdrFcShort( 0x8 ),      /* Alpha Stack size/offset = 8 */
#endif
/* 20 */        0x8,            /* FC_LONG */
                                0x0,                    /* 0 */

                /* Parameter txn */

/* 22 */        NdrFcShort( 0x201b ),   /* Flags:  must size, must free, in, out, srv alloc size=8 */
#ifndef _ALPHA_
/* 24 */        NdrFcShort( 0x8 ),      /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                NdrFcShort( 0x10 ),     /* Alpha Stack size/offset = 16 */
#endif
/* 26 */        NdrFcShort( 0x6 ),      /* Type Offset=6 */

                /* Return value */

/* 28 */        NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 30 */        NdrFcShort( 0xc ),      /* x86, MIPS, PPC Stack size/offset = 12 */
#else
                                NdrFcShort( 0x18 ),     /* Alpha Stack size/offset = 24 */
#endif
/* 32 */        0x8,            /* FC_LONG */
                                0x0,                    /* 0 */

                /* Procedure Payment */

/* 34 */        0x33,           /* FC_AUTO_HANDLE */
                                0x6c,                   /* Old Flags:  object, Oi2 */
/* 36 */        NdrFcLong( 0x0 ),       /* 0 */
/* 40 */        NdrFcShort( 0x4 ),      /* 4 */
#ifndef _ALPHA_
/* 42 */        NdrFcShort( 0x10 ),     /* x86, MIPS, PPC Stack size/offset = 16 */
#else
                                NdrFcShort( 0x20 ),     /* Alpha Stack size/offset = 32 */
#endif
/* 44 */        NdrFcShort( 0x8 ),      /* 8 */
/* 46 */        NdrFcShort( 0x10 ),     /* 16 */
/* 48 */        0x7,            /* Oi2 Flags:  srv must size, clt must size, has return, */
                                0x3,                    /* 3 */

                /* Parameter iSize */

/* 50 */        NdrFcShort( 0x158 ),    /* Flags:  in, out, base type, simple ref, */
#ifndef _ALPHA_
/* 52 */        NdrFcShort( 0x4 ),      /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                NdrFcShort( 0x8 ),      /* Alpha Stack size/offset = 8 */
#endif
/* 54 */        0x8,            /* FC_LONG */
                                0x0,                    /* 0 */

                /* Parameter txn */

/* 56 */        NdrFcShort( 0x201b ),   /* Flags:  must size, must free, in, out, srv alloc size=8 */
#ifndef _ALPHA_
/* 58 */        NdrFcShort( 0x8 ),      /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                NdrFcShort( 0x10 ),     /* Alpha Stack size/offset = 16 */
#endif
/* 60 */        NdrFcShort( 0x6 ),      /* Type Offset=6 */

                /* Return value */

/* 62 */        NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 64 */        NdrFcShort( 0xc ),      /* x86, MIPS, PPC Stack size/offset = 12 */
#else
                                NdrFcShort( 0x18 ),     /* Alpha Stack size/offset = 24 */
#endif
/* 66 */        0x8,            /* FC_LONG */
                                0x0,                    /* 0 */

                /* Procedure Delivery */

/* 68 */        0x33,           /* FC_AUTO_HANDLE */
                                0x6c,                   /* Old Flags:  object, Oi2 */
/* 70 */        NdrFcLong( 0x0 ),       /* 0 */
/* 74 */        NdrFcShort( 0x5 ),      /* 5 */
#ifndef _ALPHA_
/* 76 */        NdrFcShort( 0x10 ),     /* x86, MIPS, PPC Stack size/offset = 16 */
#else
                                NdrFcShort( 0x20 ),     /* Alpha Stack size/offset = 32 */
#endif
/* 78 */        NdrFcShort( 0x8 ),      /* 8 */
/* 80 */        NdrFcShort( 0x8 ),      /* 8 */
/* 82 */        0x6,            /* Oi2 Flags:  clt must size, has return, */
                                0x3,                    /* 3 */

                /* Parameter iSize */

/* 84 */        NdrFcShort( 0x148 ),    /* Flags:  in, base type, simple ref, */
#ifndef _ALPHA_
/* 86 */        NdrFcShort( 0x4 ),      /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                NdrFcShort( 0x8 ),      /* Alpha Stack size/offset = 8 */
#endif
/* 88 */        0x8,            /* FC_LONG */
                                0x0,                    /* 0 */

                /* Parameter txn */

/* 90 */        NdrFcShort( 0x200b ),   /* Flags:  must size, must free, in, srv alloc size=8 */
#ifndef _ALPHA_
/* 92 */        NdrFcShort( 0x8 ),      /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                NdrFcShort( 0x10 ),     /* Alpha Stack size/offset = 16 */
#endif
```

APPENDIX A - APPLICATION SOURCE CODE

```
/* 94 */        NdrFcShort( 0x18 ),         /* Type Offset=24 */

                /* Return value */

/* 96 */        NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 98 */        NdrFcShort( 0xc ),          /* x86, MIPS, PPC Stack size/offset = 12 */
#else
                                            NdrFcShort( 0x18 ),         /* Alpha Stack size/offset = 24 */
#endif
/* 100 */       0x8,                        /* FC_LONG */
                                            0x0,                        /* 0 */

                /* Procedure StockLevel */

/* 102 */       0x33,                       /* FC_AUTO_HANDLE */
                                            0x6c,                       /* Old Flags:  object, Oi2 */
/* 104 */       NdrFcLong( 0x0 ),           /* 0 */
/* 108 */       NdrFcShort( 0x6 ),          /* 6 */
#ifndef _ALPHA_
/* 110 */       NdrFcShort( 0x10 ),         /* x86, MIPS, PPC Stack size/offset = 16 */
#else
                                            NdrFcShort( 0x20 ),         /* Alpha Stack size/offset = 32 */
#endif
/* 112 */       NdrFcShort( 0x8 ),          /* 8 */
/* 114 */       NdrFcShort( 0x10 ),         /* 16 */
/* 116 */       0x7,                        /* Oi2 Flags:  srv must size, clt must size, has return, */
                                            0x3,                        /* 3 */

                /* Parameter iSize */

/* 118 */       NdrFcShort( 0x158 ),        /* Flags:  in, out, base type, simple ref, */
#ifndef _ALPHA_
/* 120 */       NdrFcShort( 0x4 ),          /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                            NdrFcShort( 0x8 ),          /* Alpha Stack size/offset = 8 */
#endif
/* 122 */       0x8,                        /* FC_LONG */
                                            0x0,                        /* 0 */

                /* Parameter txn */

/* 124 */       NdrFcShort( 0x201b ),       /* Flags:  must size, must free, in, out, srv alloc size=8 */
#ifndef _ALPHA_
/* 126 */       NdrFcShort( 0x8 ),          /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                            NdrFcShort( 0x10 ),         /* Alpha Stack size/offset = 16 */
#endif
/* 128 */       NdrFcShort( 0x6 ),          /* Type Offset=6 */

                /* Return value */

/* 130 */       NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 132 */       NdrFcShort( 0xc ),          /* x86, MIPS, PPC Stack size/offset = 12 */
#else
                                            NdrFcShort( 0x18 ),         /* Alpha Stack size/offset = 24 */
#endif
/* 134 */       0x8,                        /* FC_LONG */
                                            0x0,                        /* 0 */

                /* Procedure OrderStatus */

/* 136 */       0x33,                       /* FC_AUTO_HANDLE */
                                            0x6c,                       /* Old Flags:  object, Oi2 */
/* 138 */       NdrFcLong( 0x0 ),           /* 0 */
/* 142 */       NdrFcShort( 0x7 ),          /* 7 */
#ifndef _ALPHA_
/* 144 */       NdrFcShort( 0x10 ),         /* x86, MIPS, PPC Stack size/offset = 16 */
#else
                                            NdrFcShort( 0x20 ),         /* Alpha Stack size/offset = 32 */
#endif
/* 146 */       NdrFcShort( 0x8 ),          /* 8 */
/* 148 */       NdrFcShort( 0x10 ),         /* 16 */
/* 150 */       0x7,                        /* Oi2 Flags:  srv must size, clt must size, has return, */
                                            0x3,                        /* 3 */

                /* Parameter iSize */

/* 152 */       NdrFcShort( 0x158 ),        /* Flags:  in, out, base type, simple ref, */
#ifndef _ALPHA_
/* 154 */       NdrFcShort( 0x4 ),          /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                            NdrFcShort( 0x8 ),          /* Alpha Stack size/offset = 8 */
#endif
/* 156 */       0x8,                        /* FC_LONG */
                                            0x0,                        /* 0 */

                /* Parameter txn */

/* 158 */       NdrFcShort( 0x201b ),       /* Flags:  must size, must free, in, out, srv alloc size=8 */
#ifndef _ALPHA_
/* 160 */       NdrFcShort( 0x8 ),          /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                            NdrFcShort( 0x10 ),         /* Alpha Stack size/offset = 16 */
#endif
/* 162 */       NdrFcShort( 0x6 ),          /* Type Offset=6 */

                /* Return value */

/* 164 */       NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 166 */       NdrFcShort( 0xc ),          /* x86, MIPS, PPC Stack size/offset = 12 */
#else
                                            NdrFcShort( 0x18 ),         /* Alpha Stack size/offset = 24 */
#endif
/* 168 */       0x8,                        /* FC_LONG */
                                            0x0,                        /* 0 */

                /* Procedure CallSetComplete */
```

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

                /* Return value */

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

                                    0x0
        }
    };

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
    {
        0,
        {
                                    NdrFcShort( 0x0 ),          /* 0 */
/*  2 */
                                    0x11, 0x8,      /* FC_RP [simple_pointer] */
/*  4 */        0x8,                /* FC_LONG */
                                    0x5c,               /* FC_PAD */
/*  6 */
                                    0x11, 0x14,     /* FC_RP [alloced_on_stack] [pointer_deref] */
/*  8 */        NdrFcShort( 0x2 ),  /* Offset= 2 (10) */
/* 10 */
                                    0x13, 0x0,      /* FC_OP */
/* 12 */        NdrFcShort( 0x2 ),  /* Offset= 2 (14) */
/* 14 */
                                    0x1b,               /* FC_CARRAY */
                                    0x0,                /* 0 */
/* 16 */        NdrFcShort( 0x1 ),  /* 1 */
/* 18 */        0x28,               /* Corr desc:  parameter, FC_LONG */
                                    0x54,               /* FC_DEREFERENCE */
#ifndef _ALPHA_
/* 20 */        NdrFcShort( 0x4 ),  /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                    NdrFcShort( 0x8 ),          /* Alpha Stack size/offset = 8 */
#endif
/* 22 */        0x2,                /* FC_CHAR */
                                    0x5b,               /* FC_END */
/* 24 */
                                    0x11, 0x14,     /* FC_RP [alloced_on_stack] [pointer_deref] */
/* 26 */        NdrFcShort( 0x2 ),  /* Offset= 2 (28) */
/* 28 */
                                    0x12, 0x0,      /* FC_UP */
/* 30 */        NdrFcShort( 0xffffffd0 ),   /* Offset= -16 (14) */

                                    0x0
        }
    };

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

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

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


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

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

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

    return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) & _tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) & _tpcc_com_ps_StubVtblList,
    (const PCInterfaceName * ) & _tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    & _tpcc_com_ps_IID_Lookup,
    1,
```

```
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0  /* Filler3 */
};


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


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

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


 /* File created by MIDL compiler version 5.02.0235 */
/* at Fri Aug 13 18:56:18 1999
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
        DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

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


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


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


#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE   33
#define PROC_FORMAT_STRING_SIZE   193
#define TRANSMIT_AS_TABLE_SIZE    0
#define WIRE_MARSHAL_TABLE_SIZE   0

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

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


extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;


/* Object interface: IUnknown, ver. 0.0,
   GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */


/* Object interface: ITPCC, ver. 0.0,
   GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,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
};


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,
    0x50200eb, /* MIDL Version 5.2.235 */
    0,
    0,
    0,  /* notify & notify_flag routine table */
    1,  /* Flags */
    0,  /* Reserved3 */
    0,  /* Reserved4 */
    0   /* Reserved5 */
    };

#pragma data_seg(".rdata")

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

static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
    {
        0,
        {

                    /* Procedure NewOrder */

                                                0x33,               /* FC_AUTO_HANDLE */
                                                0x6c,               /* Old Flags:  object, Oi2 */
/*  2 */        NdrFcLong( 0x0 ),               /* 0 */
/*  6 */        NdrFcShort( 0x3 ),              /* 3 */
/*  8 */        NdrFcShort( 0x20 ),             /* ia64, axp64 Stack size/offset = 32 */
/* 10 */        NdrFcShort( 0x8 ),              /* 8 */
/* 12 */        NdrFcShort( 0x10 ),             /* 16 */
/* 14 */        0x7,                            /* Oi2 Flags:  srv must size, clt must size, has return, */
                                                0x3,                /* 3 */

                    /* Parameter iSize */

/* 16 */        NdrFcShort( 0x158 ),            /* Flags:  in, out, base type, simple ref, */
/* 18 */        NdrFcShort( 0x8 ),              /* ia64, axp64 Stack size/offset = 8 */
/* 20 */        0x8,                            /* FC_LONG */
                                                0x0,                /* 0 */

                    /* Parameter txn */

/* 22 */        NdrFcShort( 0x201b ),           /* Flags:  must size, must free, in, out, srv alloc size=8 */
/* 24 */        NdrFcShort( 0x10 ),             /* ia64, axp64 Stack size/offset = 16 */
/* 26 */        NdrFcShort( 0x6 ),              /* Type Offset=6 */

                    /* Return value */

/* 28 */        NdrFcShort( 0x70 ),             /* Flags:  out, return, base type, */
/* 30 */        NdrFcShort( 0x18 ),             /* ia64, axp64 Stack size/offset = 24 */
/* 32 */        0x8,                            /* FC_LONG */
                                                0x0,                /* 0 */

                    /* Procedure Payment */

/* 34 */        0x33,                           /* FC_AUTO_HANDLE */
                                                0x6c,               /* Old Flags:  object, Oi2 */
/* 36 */        NdrFcLong( 0x0 ),               /* 0 */
/* 40 */        NdrFcShort( 0x4 ),              /* 4 */
/* 42 */        NdrFcShort( 0x20 ),             /* ia64, axp64 Stack size/offset = 32 */
/* 44 */        NdrFcShort( 0x8 ),              /* 8 */
/* 46 */        NdrFcShort( 0x10 ),             /* 16 */
/* 48 */        0x7,                            /* Oi2 Flags:  srv must size, clt must size, has return, */
                                                0x3,                /* 3 */

                    /* Parameter iSize */
```

```
/* 50 */    NdrFcShort( 0x158 ),    /* Flags:  in, out, base type, simple ref, */
/* 52 */    NdrFcShort( 0x8 ),      /* ia64, axp64 Stack size/offset = 8 */
/* 54 */    0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Parameter txn */

/* 56 */    NdrFcShort( 0x201b ),   /* Flags:  must size, must free, in, out, srv alloc size=8 */
/* 58 */    NdrFcShort( 0x10 ),     /* ia64, axp64 Stack size/offset = 16 */
/* 60 */    NdrFcShort( 0x6 ),      /* Type Offset=6 */

            /* Return value */

/* 62 */    NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
/* 64 */    NdrFcShort( 0x18 ),     /* ia64, axp64 Stack size/offset = 24 */
/* 66 */    0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Procedure Delivery */

/* 68 */    0x33,                   /* FC_AUTO_HANDLE */
                                    0x6c,                   /* Old Flags:  object, Oi2 */
/* 70 */    NdrFcLong( 0x0 ),       /* 0 */
/* 74 */    NdrFcShort( 0x5 ),      /* 5 */
/* 76 */    NdrFcShort( 0x20 ),     /* ia64, axp64 Stack size/offset = 32 */
/* 78 */    NdrFcShort( 0x8 ),      /* 8 */
/* 80 */    NdrFcShort( 0x8 ),      /* 8 */
/* 82 */    0x6,                    /* Oi2 Flags:  clt must size, has return, */
                                    0x3,                    /* 3 */

            /* Parameter iSize */

/* 84 */    NdrFcShort( 0x148 ),    /* Flags:  in, base type, simple ref, */
/* 86 */    NdrFcShort( 0x8 ),      /* ia64, axp64 Stack size/offset = 8 */
/* 88 */    0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Parameter txn */

/* 90 */    NdrFcShort( 0x200b ),   /* Flags:  must size, must free, in, srv alloc size=8 */
/* 92 */    NdrFcShort( 0x10 ),     /* ia64, axp64 Stack size/offset = 16 */
/* 94 */    NdrFcShort( 0x18 ),     /* Type Offset=24 */

            /* Return value */

/* 96 */    NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
/* 98 */    NdrFcShort( 0x18 ),     /* ia64, axp64 Stack size/offset = 24 */
/* 100 */   0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Procedure StockLevel */

/* 102 */   0x33,                   /* FC_AUTO_HANDLE */
                                    0x6c,                   /* Old Flags:  object, Oi2 */
/* 104 */   NdrFcLong( 0x0 ),       /* 0 */
/* 108 */   NdrFcShort( 0x6 ),      /* 6 */
/* 110 */   NdrFcShort( 0x20 ),     /* ia64, axp64 Stack size/offset = 32 */
/* 112 */   NdrFcShort( 0x8 ),      /* 8 */
/* 114 */   NdrFcShort( 0x10 ),     /* 16 */
/* 116 */   0x7,                    /* Oi2 Flags:  srv must size, clt must size, has return, */
                                    0x3,                    /* 3 */

            /* Parameter iSize */

/* 118 */   NdrFcShort( 0x158 ),    /* Flags:  in, out, base type, simple ref, */
/* 120 */   NdrFcShort( 0x8 ),      /* ia64, axp64 Stack size/offset = 8 */
/* 122 */   0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Parameter txn */

/* 124 */   NdrFcShort( 0x201b ),   /* Flags:  must size, must free, in, out, srv alloc size=8 */
/* 126 */   NdrFcShort( 0x10 ),     /* ia64, axp64 Stack size/offset = 16 */
/* 128 */   NdrFcShort( 0x6 ),      /* Type Offset=6 */

            /* Return value */

/* 130 */   NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
/* 132 */   NdrFcShort( 0x18 ),     /* ia64, axp64 Stack size/offset = 24 */
/* 134 */   0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Procedure OrderStatus */

/* 136 */   0x33,                   /* FC_AUTO_HANDLE */
                                    0x6c,                   /* Old Flags:  object, Oi2 */
/* 138 */   NdrFcLong( 0x0 ),       /* 0 */
/* 142 */   NdrFcShort( 0x7 ),      /* 7 */
/* 144 */   NdrFcShort( 0x20 ),     /* ia64, axp64 Stack size/offset = 32 */
/* 146 */   NdrFcShort( 0x8 ),      /* 8 */
/* 148 */   NdrFcShort( 0x10 ),     /* 16 */
/* 150 */   0x7,                    /* Oi2 Flags:  srv must size, clt must size, has return, */
                                    0x3,                    /* 3 */

            /* Parameter iSize */

/* 152 */   NdrFcShort( 0x158 ),    /* Flags:  in, out, base type, simple ref, */
/* 154 */   NdrFcShort( 0x8 ),      /* ia64, axp64 Stack size/offset = 8 */
/* 156 */   0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Parameter txn */

/* 158 */   NdrFcShort( 0x201b ),   /* Flags:  must size, must free, in, out, srv alloc size=8 */
/* 160 */   NdrFcShort( 0x10 ),     /* ia64, axp64 Stack size/offset = 16 */
/* 162 */   NdrFcShort( 0x6 ),      /* Type Offset=6 */

            /* Return value */

/* 164 */   NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
```

APPENDIX A - APPLICATION SOURCE CODE

```
/* 166 */      NdrFcShort( 0x18 ),      /* ia64, axp64 Stack size/offset = 24 */
/* 168 */      0x8,                     /* FC_LONG */
                                        0x0,                    /* 0 */

               /* Procedure CallSetComplete */

/* 170 */      0x33,                    /* FC_AUTO_HANDLE */
                                        0x6c,                   /* Old Flags:  object, Oi2 */
/* 172 */      NdrFcLong( 0x0 ),        /* 0 */
/* 176 */      NdrFcShort( 0x8 ),       /* 8 */
/* 178 */      NdrFcShort( 0x10 ),      /* ia64, axp64 Stack size/offset = 16 */
/* 180 */      NdrFcShort( 0x0 ),       /* 0 */
/* 182 */      NdrFcShort( 0x8 ),       /* 8 */
/* 184 */      0x4,                     /* Oi2 Flags:  has return, */
                                        0x1,                    /* 1 */

               /* Return value */

/* 186 */      NdrFcShort( 0x70 ),      /* Flags:  out, return, base type, */
/* 188 */      NdrFcShort( 0x8 ),       /* ia64, axp64 Stack size/offset = 8 */
/* 190 */      0x8,                     /* FC_LONG */
                                        0x0,                    /* 0 */

                                        0x0
    }
  };

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
  {
    0,
    {
                                        NdrFcShort( 0x0 ),      /* 0 */
/*  2 */
                                        0x11, 0x8,      /* FC_RP [simple_pointer] */
/*  4 */      0x8,                     /* FC_LONG */
                                        0x5c,                   /* FC_PAD */
/*  6 */
                                        0x11, 0x14,     /* FC_RP [alloced_on_stack] [pointer_deref] */
/*  8 */      NdrFcShort( 0x2 ),       /* Offset= 2 (10) */
/* 10 */
                                        0x13, 0x0,      /* FC_OP */
/* 12 */      NdrFcShort( 0x2 ),       /* Offset= 2 (14) */
/* 14 */
                                        0x1b,                   /* FC_CARRAY */
                                        0x0,                    /* 0 */
/* 16 */      NdrFcShort( 0x1 ),       /* 1 */
/* 18 */      0x28,                    /* Corr desc:  parameter, FC_LONG */
                                        0x54,                   /* FC_DEREFERENCE */
/* 20 */      NdrFcShort( 0x8 ),       /* ia64, axp64 Stack size/offset = 8 */
/* 22 */      0x2,                     /* FC_CHAR */
                                        0x5b,                   /* FC_END */
/* 24 */
                                        0x11, 0x14,     /* FC_RP [alloced_on_stack] [pointer_deref] */
/* 26 */      NdrFcShort( 0x2 ),       /* Offset= 2 (28) */
/* 28 */
                                        0x12, 0x0,      /* FC_UP */
/* 30 */      NdrFcShort( 0xffffffff0 ),/* Offset= -16 (14) */

                                        0x0
    }
  };

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

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

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


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

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

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

  return 0;
}

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

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

tpcc_common_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 Design

B.1 Create, backup and restore

SETUP.CMD

```
::@ECHO OFF

@ECHO ********************************************************************************
@ECHO *                                                                              *
@ECHO *  Microsoft TPC-C V3 Benchmark Kit Ver. 4.20                                   *
@ECHO *                                                                              *
@ECHO ********************************************************************************

@if '%1'==''  goto usage
@if '%2'==''  goto usage
@if '%3'==''  goto usage
@if not '%4'=='' if not '%4' == 'normal' if not '%4' == 'scale_down' goto usage

:: Cleanup any old .err files
@@if exist logs\*.err del logs\*.err                                                    >nul

@if  '%3'=='full' goto start
@if  '%3'=='builddb' goto builddb
@if  '%3'=='objects' goto objects
@if  '%3'=='bulkload' goto bulkload
@if  '%3'=='objectsfull' goto objects
@if  '%3'=='bulkloadfull' goto bulkload
@if  '%3'=='backup' goto backup
goto usage

:start
:: Cleanup the logs directory...
@if exist logs\version.log del logs\version.log                                         >nul
@if exist logs\db.log del logs\db.log                                                   >nul
@if exist logs\objects.log del logs\objects.log                                         >nul
@if exist logs\objects.log del logs\objects.log                                         >nul
@if exist logs\bulkload.log del logs\bulkload.log                                       >nul
@if exist logs\backup.log del logs\backup.log                                           >nul

@isql -Usa -P -S%1 -Q"select @@version"                                                 > logs\version.log
@isql -Usa -P -S%1 -Q"select getdate()"                                                 >> logs\version.log

:Verify_Installation
@isql -Usa -P -S%1 -b -iscripts\utility\verify_msg.sql                          >nul
@isql -Usa -P -S%1 -b -iscripts\utility\verify_build.sql                >nul
@isql -Usa -P -S%1 -b -iscripts\utility\verify_sort.sql                         >nul
@isql -Usa -P -S%1 -b -Q"ms_verify_build"                                               >nul
@if errorlevel 1 goto BAD_BUILD
@isql -Usa -P -S%1 -b -Q"ms_verify_sort"                                                >nul
@if errorlevel 1 goto BAD_SORT

:builddb
@@if exist logs\db.log del logs\db.log                                                  >nul
@ECHO Removing any existing TPCC database and backup devices...
@isql -Usa -P -S%1 -e < scripts\%2.war\database\removedb.sql                     >  logs\db.log
@ECHO Creating Backup Device(s)...
@isql -Usa -P -S%1 -e < scripts\%2.war\database\backupdev.sql                    >> logs\db.log
@if errorlevel 1 goto CREATE_ERROR
@ECHO Building database files and database...
@isql -Usa -P -S%1 -b -e < scripts\%2.war\database\createdb.sql                  >> logs\db.log
@if errorlevel 1 goto CREATE_ERROR
@ECHO Database build complete.
@if  '%3'=='full' goto objects
goto end

:objects
@if exist logs\objects.log del logs\objects.log                                         >nul
@ECHO Creating TPC-C database tables...
@isql -Usa -P -S%1 -b -e < scripts\%2.war\ddl\tables.sql                         >  logs\objects.log
@if errorlevel 1 goto TABLES_ERROR
@ECHO Creating database objects...
@isql -Usa -P -S%1 -b -e < scripts\dml\neword.sql                                >> logs\objects.log
@if errorlevel 1 goto NEWORDER_ERROR
@isql -Usa -P -S%1 -b -e < scripts\dml\payment.sql                               >> logs\objects.log
@if errorlevel  1 goto PAYMENT_ERROR
@isql -Usa -P -S%1 -b -e < scripts\dml\ordstat.sql                               >> logs\objects.log
@if errorlevel 1 goto ORDERSTATUS_ERROR
@isql -Usa -P -S%1 -b -e < scripts\dml\delivery.sql                              >> logs\objects.log
@if errorlevel 1 goto DELIVERY_ERROR
@isql -Usa -P -S%1 -b -e < scripts\dml\stocklev.sql                              >> logs\objects.log
@if errorlevel 1 goto STOCKLEVEL_ERROR
@isql -Usa -P -S%1 -e < scripts\dml\version.sql                                         >> logs\objects.log
@ECHO Database object creation complete.
@if  '%3'=='full' goto bulkload
@if  '%3'=='objectsfull' goto bulkload
goto end

:bulkload
@if exist logs\bulkload.log del logs\bulkload.log                                       >nul
@ECHO Setting database options before load...
@isql -Usa -P -S%1 -b -e < scripts\utility\dbopt1.sql                            >> logs\objects.log
@if errorlevel 1 goto DBOPT1_ERROR
@ECHO Beginning data load and index creation...
@if '%4'==''      loader \%PROCESSOR_ARCHITECTURE%\bin\tpccldr -S%1 -W%2 -flogs\bulkload.log -dscripts\%2.war\ddl -c0
@if errorlevel 1 goto END
@if '%4'=='normal' loader \%PROCESSOR_ARCHITECTURE%\bin\tpccldr -S%1 -W%2 -flogs\bulkload.log -dscripts\%2.war\ddl -
c0
@if errorlevel 1 goto END
@if '%4'=='scale_down' loader \%PROCESSOR_ARCHITECTURE%\bin\tpccldr -S%1 -W%2 -flogs\bulkload.log -
dscripts\%2.war\ddl -c1
@if errorlevel 1 goto END
goto bulkloaddone
:bulkloaddone
@ECHO Setting database options after load...
```

APPENDIX B – DATABASE DESIGN

```
@isql -Usa -P -S%1 -b -e < scripts\utility\dbopt2.sql                    >> logs\bulkload.log
@if errorlevel 1 goto DBOPT2_ERROR
@ECHO Data load and index creation complete.


@if  '%3'=='full' goto backup
@if  '%3'=='objectsfull' goto backup
@if  '%3'=='bulkloadfull' goto backup
goto end

:backup
@if exist logs\backup.log del logs\backup.log                                    >nul
@ECHO Backing up database...
@isql -Usa -P -S%1 -b -e < scripts\%2.war\database\backup.sql             >  logs\backup.log
@if errorlevel 1 goto BACKUP_ERROR
@ECHO Database backup complete.
@if  '%3'=='full' goto verifyload
@if  '%3'=='objectsfull' goto verifyload
@if  '%3'=='bulkloadfull' goto verifyload
goto complete

:verifyload
@if exist logs\verifyload.log del logs\verifyload.log                            >nul
@Echo Verifying TPC-C database load...
@isql -Usa -P -S%1 -b -e < scripts\utility\verifytpccload.sql          >  logs\verifyload.log
@if errorlevel 1 goto VERIFY_ERROR
@ECHO Check logs\verifyload.log to verify database load.

:complete
@ECHO ********************************************************************************
@ECHO *                                                                              *
@ECHO *  Full TPC-C V3 build complete.  Check logs directory for setup errors.  *
@ECHO *                                                                              *
@ECHO ********************************************************************************

goto end

:usage
@ECHO ********************************************************************************
@ECHO *                                                                              *
@ECHO *  The TPC-C setup command file requires the following parameters:      *
@ECHO *                                                                              *
@ECHO *  setup SERVER NUMWAR BLDOPT VERSION DBTYPE                          *
@ECHO *                                                                              *
@ECHO *     SERVER  = machine name of server (use "" for local server)       *
@ECHO *     NUMWAR  = number of warehouses                                   *
@ECHO *     BLDOPT  = full, builddb, objects, objectsfull, bulkload,         *
@ECHO *               bulkloadfull, or backup                                *
@ECHO *     DBTYPE  = normal or scale_down                                   *
@ECHO *                                                                              *
@ECHO *  Note #1: the BLDOPT and VERSION parameters are case sensitive.       *
@ECHO *                                                                              *
@ECHO *  Note #2: the DBTYPE is optional.  If no DBTYPE is specified, SETUP   *
@ECHO *          will default to NORMAL.                                     *
@ECHO *                                                                              *
@ECHO *  Example:                                                            *
@ECHO *                                                                              *
@ECHO *  The following command would be used to build a complete 200         *
@ECHO *  warehouse database on SQL Server 7.0 runnning on server \\myserver.  *
@ECHO *                                                                              *
@ECHO *      SETUP myserver 200 full                                         *
@ECHO *                                                                              *
@ECHO *  NOTE 1: This command file does a backup of the database by default   *
@ECHO *  after the database build process is complete.  If you do not wish    *
@ECHO *  to make a backup (strongly discouraged), you must edit this file     *
@ECHO *  and comment that section out.  Also, if you need to run the dbcheck  *
@ECHO *  and the dbtables scripts on the fresh database load for an audit,    *
@ECHO *  you must either run them manually or edit this file to include them. *
@ECHO *                                                                              *
@ECHO *  NOTE 2: The TPC-C setup program supports both Intel and Alpha        *
@ECHO *  systems.  It queries the %PROCESSOR_ARCHITECTURE% environment        *
@ECHO *  variable and runs the appropriate executables.                      *
@ECHO *                                                                              *
@ECHO ********************************************************************************
@goto end


:CREATE_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error in the database/backup device creation.
@echo.
@echo Check your CREATEDB.SQL, BACKUPDEV.SQL, LOGS\DB.LOG, and the
@echo SQL Server errorlog (MSSQL7\LOG\ERRORLOG) for details.
@echo.
@goto END

:TABLES_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error in the table creation.
@echo.
@echo Verify that the FileGroup names specified in CREATEDB.SQL
@echo match those specified in SCRIPTS\DDL\TABLES.SQL.
@echo.
@goto END

:NEWORDER_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error in the creation of the New Order stored procedure.
@echo.
@echo Check your LOGS\OBJECTS.LOG, SCRIPTS\DML\NEWORD.SQL and the
@echo SQL Server errorlog (MSSQL7\LOG\ERRORLOG) for details.
@echo.
@goto END
```

```
:PAYMENT_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error in the creation of the Payment stored procedure.
@echo.
@echo Check your LOGS\OBJECTS.LOG, SCRIPTS\DML\PAYMENT.SQL and the
@echo SQL Server errorlog (MSSQL7\LOG\ERRORLOG) for details.
@echo.
@goto END

:ORDERSTATUS_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error in the creation of the Order Status stored procedure.
@echo.
@echo Check your LOGS\OBJECTS.LOG, SCRIPTS\DML\ORDSTAT.SQL and the
@echo SQL Server errorlog (MSSQL7\LOG\ERRORLOG) for details.
@echo.
@goto END

:DELIVERY_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error in the creation of the Delivery stored procedure.
@echo.
@echo Check your LOGS\OBJECTS.LOG, SCRIPTS\DML\DELIVERY.SQL and the
@echo SQL Server errorlog (MSSQL7\LOG\ERRORLOG) for details.
@echo.
@goto END

:STOCKLEVEL_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error in the creation of the Stock Level stored procedure.
@echo.
@echo Check your LOGS\OBJECTS.LOG, SCRIPTS\DML\STOCKLEV.SQL and the
@echo SQL Server errorlog (MSSQL7\LOG\ERRORLOG) for details.
@echo.
@goto END

:DBOPT1_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error setting the database options before load.
@echo.
@echo Check your LOGS\OBJECTS.LOG and the SQL Server errorlog
@echo (MSSQL7\LOG\ERRORLOG) for details.
@echo.
@goto END

:DBOPT2_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error setting the database options after load.
@echo.
@echo Check your LOGS\OBJECTS.LOG and the SQL Server errorlog
@echo (MSSQL7\LOG\ERRORLOG) for details.
@echo.
@goto END

:BACKUP_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error backing up the database after load.
@echo.
@echo Check your LOGS\BACKUP.LOG and the SQL Server errorlog
@echo (MSSQL7\LOG\ERRORLOG) for details.
@echo.
@goto END

:VERIFY_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error performing TPC-C database verification.
@echo.
@echo Check your LOGS\VERIFYLOAD.LOG and the SQL Server errorlog
@echo (MSSQL7\LOG\ERRORLOG) for details.
@echo.
@goto END

:BAD_BUILD
@echo.
@echo BUILD ABORTED!
@echo.
@echo Incorrect SQL Server Version.  You must run Microsoft SQL Server
@echo Version 7.00.623 or newer to use this kit.  Please uninstall this
@echo version and re-install an appropriate version of SQL Server.
@echo.
@goto END

:BAD_SORT
@echo.
@echo BUILD ABORTED!
@echo.
@echo Incorrect SQL Server Sort Order.  For performance and compatibility
@echo issues, you must run SQL Server with the Binary Sort Order.  Please
@echo re-install SQL Server and specify the Binary Sort Order.
@echo.
@goto END
:end

echo on
```

APPENDIX B – DATABASE DESIGN

CREATEDB.SQL

```
-- File:    CREATEDB.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates tpcc database and backup files for 10 warehouses


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                    = MSSQL80_tpcc_root,
                FILENAME    = "C:\MSSQL80_tpcc_root.mdf",
                SIZE                    = 10MB,
                FILEGROWTH              = 0),
FILEGROUP MSSQL80_misc_fg
(               NAME                    = MSSQL80_misc1,
                FILENAME    = "O:",
                SIZE                    = 16000MB,
                FILEGROWTH              = 0),
(               NAME                    = MSSQL80_misc2,
                FILENAME    = "P:",
                SIZE                    = 16000MB,
                FILEGROWTH              = 0),
(               NAME                    = MSSQL80_misc3,
                FILENAME    = "Q:",
                SIZE                    = 16000MB,
                FILEGROWTH              = 0),
(               NAME                    = MSSQL80_misc4,
                FILENAME    = "R:",
                SIZE                    = 16000MB,
                FILEGROWTH              = 0),
(               NAME                    = MSSQL80_misc5,
                FILENAME    = "S:",
                SIZE                    = 16000MB,
                FILEGROWTH              = 0),
(               NAME                    = MSSQL80_misc6,
                FILENAME    = "T:",
                SIZE                    = 16000MB,
                FILEGROWTH              = 0),
FILEGROUP MSSQL80_cs_fg
(               NAME                    = MSSQL80_cs1,
                FILENAME    = "F:",
                SIZE                    = 34000MB,
                FILEGROWTH              = 0),
(               NAME                    = MSSQL80_cs2,
                FILENAME    = "G:",
                SIZE                    = 34000MB,
                FILEGROWTH              = 0),
(               NAME                    = MSSQL80_cs3,
                FILENAME    = "H:",
                SIZE                    = 34000MB,
                FILEGROWTH              = 0),
(               NAME                    = MSSQL80_cs4,
                FILENAME    = "I:",
                SIZE                    = 34000MB,
                FILEGROWTH              = 0),
(               NAME                    = MSSQL80_cs5,
                FILENAME    = "J:",
                SIZE                    = 34000MB,
                FILEGROWTH              = 0),
(               NAME                    = MSSQL80_cs6,
                FILENAME    = "K:",
                SIZE                    = 34000MB,
                FILEGROWTH              = 0)
LOG ON
(               NAME                    =MSSQL80_tpcc_log,
                FILENAME    ="E:",
                SIZE                    =120000MB,
                FILEGROWTH              =0)
go

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

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

--              remove temporary table

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

APPENDIX B – DATABASE DESIGN

BACKUP.SQL

```
-- File:    BACKUP.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.10
--          Copyright Microsoft, 1999
-- Purpose:  Creates backup of tpcc database


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

dump database tpcc to tpccback_1, tpccback_2, tpccback_3, tpccback_4, tpccback_5 with init, stats = 5

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

go
```

BACKUPDEV.SQL

```
-- File:    BACKUPDEVB.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.10
--          Copyright Microsoft, 1999
-- Purpose:  Creates tpcc database Backup Devices


use master
go

-- create backup devices
-- these are nt5 filesystem junction points

exec sp_addumpdevice 'disk','tpccback_1','V:\tpccback_1.dmp'
exec sp_addumpdevice 'disk','tpccback_2','V:\tpccback_2.dmp'
exec sp_addumpdevice 'disk','tpccback_3','W:\tpccback_3.dmp'
exec sp_addumpdevice 'disk','tpccback_4','W:\tpccback_4.dmp'
exec sp_addumpdevice 'disk','tpccback_5','W:\tpccback_5.dmp'

go
```

RESTORE.SQL

```
-- File:    RESTORE.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.10
--          Copyright Microsoft, 1999
-- 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 tpccback_1, tpccback_2, tpccback_3, tpccback_4, tpccback_5 with stats = 5

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

go
```

B.2 Build indices

IDXCUSCL.SQL

```
-- File:    IDXCUSCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on customer table


use tpcc
go

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

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

create unique clustered index customer_c1 on customer(c_w_id, c_d_id, c_id)
                on MSSQL80_cs_fg

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

go
```

IDXCUSNC.SQL

```
-- File:    IDXCUSNC.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999,2000
-- Purpose:  Creates non-clustered index on customer table


use tpcc
go

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

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

create unique nonclustered index customer_nc1 on customer(c_w_id, c_d_id, c_last, c_first, c_id)
                on MSSQL80_cs_fg

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

go
IDXDISCL.SQL
-- File:    IDXDISCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on district table


use tpcc
go

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

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

create unique clustered index  district_c1 on district(d_w_id, d_id)
                with fillfactor=100 on MSSQL80_misc_fg

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

go
```

IDXITMCL.SQL

```
-- File:    IDXITMCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on item table


use tpcc
go
```

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

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

create unique clustered index item_c1 on item(i_id)
                on MSSQL80_misc_fg

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

go
```

IDXNODCL.SQL

```
-- File:    IDXNODCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on new_order table


use tpcc
go

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

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

create unique clustered index new_order_c1 on new_order(no_w_id, no_d_id, no_o_id)
                on MSSQL80_misc_fg

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

go
IDXODLCL.SQL
-- File:    IDXODLCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on order_line table
```

```
use tpcc
go

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

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

create unique clustered index order_line_c1 on order_line(ol_w_id, ol_d_id, ol_o_id, ol_number)
                on MSSQL80_misc_fg

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

go
```

IDXORDCL.SQL

```
-- File:    IDXORDCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on orders table


use tpcc
go

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

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

create unique clustered index orders_c1 on orders(o_w_id, o_d_id, o_id)
                on MSSQL80_misc_fg

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

go
```

IDXORDNC.SQL

```
-- File:    IDXORDNC.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates non-clustered index on orders table


use tpcc
go

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

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

create index orders_nc1 on orders(o_w_id, o_d_id, o_c_id, o_id)
                on MSSQL80_misc_fg

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

go
```

IDXSTKCL.SQL

```
-- File:    IDXSTKCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on stock table


use tpcc
go

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

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

create unique clustered index stock_c1 on stock(s_i_id, s_w_id)
                on MSSQL80_cs_fg

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

go
```

IDXWARCL.SQL

```
-- File:    IDXWARCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates clustered index on warehouse table


use tpcc
go

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

if exists ( select name from sysindexes where name = 'warehouse_c1' )
                drop index warehouse.warehouse_c1

create unique clustered index warehouse_c1 on warehouse(w_id)
                with fillfactor=100 on MSSQL80_misc_fg

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

go
```

B.3 Database Options

DBOPT1.SQL

```
-- File:    DBOPT1.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Sets database options for data load


use master
go
```

```
exec sp_dboption tpcc,'select into/bulkcopy',true
exec sp_dboption tpcc,'trunc. log on chkpt.',true
go

use tpcc
go

checkpoint
go
```

DBOPT2.SQL

```
-- File:    DBOPT2.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Resets database options after data load

sp_dboption tpcc,'select into/bulkcopy',FALSE
GO

sp_dboption tpcc,'trunc. log on chkpt.',FALSE
GO

USE tpcc
GO

CHECKPOINT
GO

sp_configure 'allow updates',1
GO

RECONFIGURE WITH OVERRIDE
GO

DECLARE    @msg                    varchar(50)

IF (SELECT (SUBSTRING((SELECT @@version),1,25))) = 'Microsoft SQL Server 2000'
            BEGIN
                        --                                  --
                        --      OPTIONS FOR SQL SERVER 8.0      --
                        -- Set option values for user-defined indexes --
                        --                                  --

                        SET         @msg        = ' '
                        PRINT       @msg                    --
                        SET         @msg        = 'Setting SQL Server 8.0 indexoptions'
                        PRINT       @msg
                        SET         @msg        = ' '
                        PRINT       @msg                    --

                        EXEC sp_indexoption     'customer',     'DisAllowPageLocks',        TRUE
                        EXEC sp_indexoption     'district',     'DisAllowPageLocks',        TRUE
                        EXEC sp_indexoption     'warehouse',    'DisAllowPageLocks',        TRUE
                        EXEC sp_indexoption     'stock',        'DisAllowPageLocks',        TRUE
                        EXEC sp_indexoption     'order_line',   'DisAllowRowLocks',         TRUE
                        EXEC sp_indexoption     'orders',       'DisAllowRowLocks',         TRUE
                        EXEC sp_indexoption     'new_order',    'DisAllowRowLocks',         TRUE
                        EXEC sp_indexoption     'item',                 'DisAllowRowLocks',     TRUE
                        EXEC sp_indexoption     'item',                 'DisAllowPageLocks',    TRUE
            END
ELSE
            BEGIN
                        --                                  --
                        --      OPTIONS FOR SQL SERVER 7.0      --
                        -- Set option values for user-defined indexes --
                        --

                        SET         @msg        = ' '
                        PRINT       @msg                    --
                        SET         @msg        = 'Setting SQL Server 7.0 indexoptions'
                        PRINT       @msg
                        SET         @msg        = ' '
                        PRINT       @msg                    --

                        EXEC sp_indexoption     'customer',     'AllowPageLocks',           FALSE
                        EXEC sp_indexoption     'district',     'AllowPageLocks',           FALSE
                        EXEC sp_indexoption     'warehouse',    'AllowPageLocks',           FALSE
                        EXEC sp_indexoption     'stock',        'AllowPageLocks',           FALSE
                        EXEC sp_indexoption     'order_line',   'AllowRowLocks',            FALSE
                        EXEC sp_indexoption     'orders',       'AllowRowLocks',            FALSE
                        EXEC sp_indexoption     'new_order',    'AllowRowLocks',            FALSE
                        EXEC sp_indexoption     'item',                 'AllowRowLocks',        FALSE
                        EXEC sp_indexoption     'item',                 'AllowPageLocks',       FALSE

            END
GO

Print ' '
Print '******************'
Print 'Pre-specified Locking Hierarchy:'
Print '            Lockflag = 0 ==> No pre-specified hierarchy'
Print '   Lockflag = 1 ==> Lock at Page-level then Table-level'
Print '   Lockflag = 2 ==> Lock at Row-level then Table-level'
Print '   Lockflag = 3 ==> Lock at Table-level'
Print ' '

SELECT      name,lockflags
FROM        sysindexes
WHERE       object_id('warehouse')      = id OR
            object_id('district')       = id OR
            object_id('customer')       = id OR
            object_id('stock')          = id OR
            object_id('orders')         = id OR
            object_id('order_line')     = id OR
```

```
                object_id('history')            = id OR
                object_id('new_order')          = id OR
                object_id('item')               = id
ORDER           BY lockflags asc
GO

sp_configure 'allow updates',0
GO

RECONFIGURE WITH OVERRIDE
GO

EXEC sp_dboption tpcc,      'auto update statistics',       FALSE
EXEC sp_dboption tpcc,      'auto create statistics',       FALSE
GO

EXEC sp_tableoption         'district',     'pintable',true
EXEC sp_tableoption         'warehouse',    'pintable',true
EXEC sp_tableoption         'new_order',    'pintable',true
EXEC sp_tableoption         'item',                         'pintable',true
GO
```

B.4 Table definitions

table.SQL

```
-- File:    TABLES.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates TPC-C tables


use tpcc
go

--
-- Remove all existing TPC-C tables
--

if exists ( select name from sysobjects where name = 'warehouse' )
                drop table warehouse
go
if exists ( select name from sysobjects where name = 'district' )
                drop table district
go
if exists ( select name from sysobjects where name = 'customer' )
                drop table customer
go
if exists ( select name from sysobjects where name = 'history' )
                drop table history
go
if exists ( select name from sysobjects where name = 'new_order' )
                drop table new_order
go
if exists ( select name from sysobjects where name = 'orders' )
                drop table orders
go
if exists ( select name from sysobjects where name = 'order_line' )
                drop table order_line
go
if exists ( select name from sysobjects where name = 'item' )
                drop table item
go
if exists ( select name from sysobjects where name = 'stock' )
                drop table stock
go

--
-- Create new tables
--

create table warehouse
(
                w_id                                    smallint,
                w_name                                  char(10),
                w_street_1                      char(20),
                w_street_2                      char(20),
                w_city                                  char(20),
                w_state                                 char(2),
                w_zip                                   char(9),
                w_tax                                   numeric(4,4),
                w_ytd                                   numeric(12,2)
) on MSSQL80_misc_fg
go

create table district
(
                d_id                                    tinyint,
                d_w_id                                  smallint,
                d_name                                  char(10),
                d_street_1                      char(20),
                d_street_2                      char(20),
                d_city                                  char(20),
                d_state                                 char(2),
                d_zip                                   char(9),
                d_tax                                   numeric(4,4),
                d_ytd                                   numeric(12,2),
                d_next_o_id                     int
) on MSSQL80_misc_fg
go

create table customer
(
```

```
        c_id                    int,
        c_d_id                  tinyint,
        c_w_id                  smallint,
        c_first                 char(16),
        c_middle                char(2),
        c_last                  char(16),
        c_street_1              char(20),
        c_street_2              char(20),
        c_city                  char(20),
        c_state                 char(2),
        c_zip                   char(9),
        c_phone                 char(16),
        c_since                 datetime,
        c_credit                char(2),
        c_credit_lim    numeric(12,2),
        c_discount              numeric(4,4),
        c_balance               numeric(12,2),
        c_ytd_payment           numeric(12,2),
        c_payment_cnt           smallint,
        c_delivery_cnt          smallint,
        c_data                  char(500)
) on MSSQL80_cs_fg
go

create table history
(
        h_c_id                  int,
        h_c_d_id                tinyint,
        h_c_w_id                smallint,
        h_d_id                  tinyint,
        h_w_id                  smallint,
        h_date                  datetime,
        h_amount                numeric(6,2),
        h_data                  char(24)
) on MSSQL80_misc_fg
go

create table new_order
(
        no_o_id                 int,
        no_d_id                 tinyint,
        no_w_id                 smallint
) on MSSQL80_misc_fg
go

create table orders
(
        o_id                    int,
        o_d_id                  tinyint,
        o_w_id                  smallint,
        o_c_id                  int,
        o_entry_d               datetime,
        o_carrier_id    tinyint,
        o_ol_cnt                tinyint,
        o_all_local             tinyint
) on MSSQL80_misc_fg
go

create table order_line
(
        ol_o_id                 int,
        ol_d_id                 tinyint,
        ol_w_id                 smallint,
        ol_number               tinyint,
        ol_i_id                 int,
        ol_supply_w_id          smallint,
        ol_delivery_d   datetime,
        ol_quantity             smallint,
        ol_amount               numeric(6,2),
        ol_dist_info    char(24)
) on MSSQL80_misc_fg
go

create table item
(
        i_id                    int,
        i_im_id                 int,
        i_name                  char(24),
        i_price                 numeric(5,2),
        i_data                  char(50)
) on MSSQL80_misc_fg
go

create table stock
(
        s_i_id                  int,
        s_w_id                  smallint,
        s_quantity              smallint,
        s_dist_01               char(24),
        s_dist_02               char(24),
        s_dist_03               char(24),
        s_dist_04               char(24),
        s_dist_05               char(24),
        s_dist_06               char(24),
        s_dist_07               char(24),
        s_dist_08               char(24),
        s_dist_09               char(24),
        s_dist_10               char(24),
        s_ytd                   int,
        s_order_cnt             smallint,
        s_remote_cnt            smallint,
        s_data                  char(50)
) on MSSQL80_cs_fg
go
```

B.5 Stored Procedures

APPENDIX B – DATABASE DESIGN

NEWORD.SQL

```
-- File:    NEWORD.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21.000
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates new order transaction stored procedure
--
--                  Interface Level: 4.10.000

use tpcc
go

if exists ( select name from sysobjects where name = "tpcc_neworder" )
                drop procedure tpcc_neworder
go

create proc tpcc_neworder
                                                @w_id           smallint,
                                                @d_id           tinyint,
                                                @c_id           int,
                                                @o_ol_cnt       tinyint,
                                                @o_all_local    tinyint,
                                                @i_id1  int = 0, @s_w_id1  smallint = 0, @ol_qty1  smallint =
0,
                                                @i_id2  int = 0, @s_w_id2  smallint = 0, @ol_qty2  smallint =
0,
                                                @i_id3  int = 0, @s_w_id3  smallint = 0, @ol_qty3  smallint =
0,
                                                @i_id4  int = 0, @s_w_id4  smallint = 0, @ol_qty4  smallint =
0,
                                                @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
```

DELIVERY.SQL

```
-- File:    DELIVERY.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21.000
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates delivery transaction stored procedure
--
--              Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_delivery" )
        drop procedure tpcc_delivery
go

create proc tpcc_delivery    @w_id        smallint,
                                                @o_carrier_id   smallint
as

declare @d_id               tinyint,
    @o_id   int,
    @c_id   int,
    @total  numeric(12,2),
    @oid1   int,
    @oid2   int,
    @oid3   int,
    @oid4   int,
    @oid5   int,
    @oid6   int,
    @oid7   int,
    @oid8   int,
    @oid9   int,
    @oid10  int
```

```
select @d_id = 0

begin tran d

        while (@d_id < 10)
        begin

            select  @d_id  = @d_id + 1,
                    @total = 0,
                    @o_id  = 0

                select  top 1
                        @o_id       = no_o_id
                from    new_order (serializable updlock)
                where   no_w_id     = @w_id and
                        no_d_id     = @d_id
                order   by no_o_id asc

            if (@@rowcount <> 0)
                begin

-- claim the order for this district

                        delete  new_order
                        where   no_w_id     = @w_id and
                                no_d_id     = @d_id and
                                no_o_id     = @o_id

-- set carrier_id on this order (and get customer id)

                        update  orders
                        set     o_carrier_id  = @o_carrier_id,
                                @c_id                   = o_c_id
                        where   o_w_id                  = @w_id and
                                o_d_id                  = @d_id and
                                o_id                    = @o_id

-- set date in all lineitems for this order (and sum amounts)

                        update  order_line
                        set     ol_delivery_d  = getdate(),
                                @total                  = @total + ol_amount
                        where   ol_w_id                 = @w_id and
                                ol_d_id                 = @d_id and
                                ol_o_id                 = @o_id

-- accummulate lineitem amounts for this order into customer

                        update  customer
                        set     c_balance      = c_balance + @total,
                                c_delivery_cnt          = c_delivery_cnt + 1
                        where   c_w_id                  = @w_id and
                                c_d_id                  = @d_id and
                                c_id                    = @c_id

                end

                select @oid1 = case @d_id when  1  then @o_id else @oid1 end,
                @oid2 = case @d_id when  2  then @o_id else @oid2 end,
                @oid3 = case @d_id when  3  then @o_id else @oid3 end,
                @oid4 = case @d_id when  4  then @o_id else @oid4 end,
                @oid5 = case @d_id when  5  then @o_id else @oid5 end,
                @oid6 = case @d_id when  6  then @o_id else @oid6 end,
                @oid7 = case @d_id when  7  then @o_id else @oid7 end,
                @oid8 = case @d_id when  8  then @o_id else @oid8 end,
                @oid9 = case @d_id when  9  then @o_id else @oid9 end,
                @oid10 = case @d_id when 10 then @o_id else @oid10 end

        end

commit tran d

-- return delivery data to client

select @oid1,
    @oid2,
    @oid3,
    @oid4,
    @oid5,
    @oid6,
    @oid7,
    @oid8,
    @oid9,
    @oid10

go
```

ORDSTAT.SQL

```
-- File:    ORDSTAT.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21.000
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates order status transaction stored procedure
--
--              Interface Level: 4.10.000

use tpcc
go

if exists ( select name from sysobjects where name = "tpcc_orderstatus" )
            drop procedure   tpcc_orderstatus
go

create proc tpcc_orderstatus            @w_id       smallint,
                                        @d_id       tinyint,
```

```
                                        @c_id       int,
                                        @c_last     char(16) = ""
as

declare @c_balance              numeric(12,2),
        @c_first                char(16),
        @c_middle               char(2),
        @o_id                   int,
        @o_entry_d              datetime,
        @o_carrier_id           smallint,
        @cnt                    smallint

begin tran o

if (@c_id = 0)
                begin

-- get customer id and info using last name

                        select  @cnt            = (count(*)+1)/2
                        from    customer (repeatableread)
                        where   c_last          = @c_last and
                                c_w_id          = @w_id and
                                c_d_id          = @d_id

                        set     rowcount @cnt

                        select  @c_id                   = c_id,
                                @c_balance      = c_balance,
                                  @c_first      = c_first,
                                  @c_last       = c_last,
                                  @c_middle     = c_middle
                        from    customer (repeatableread)
                        where   c_last                  = @c_last and
                                c_w_id                  = @w_id and
                                c_d_id                  = @d_id
                        order   by c_w_id, c_d_id, c_last, c_first

                        set     rowcount 0
                end

                else

                begin

-- get customer info if by id

                        select  @c_balance      = c_balance,
                                @c_first        = c_first,
                                @c_middle       = c_middle,
                                @c_last                 = c_last
                        from    customer (repeatableread)
                        where   c_id                    = @c_id and
                                c_d_id                  = @d_id and
                                c_w_id                  = @w_id

                        select  @cnt            = @@rowcount

                end

-- if no such customer

                if (@cnt = 0)
                begin
                        raiserror("Customer not found",18,1)
                        goto custnotfound
                end

-- get order info

                select  @o_id                   = o_id,
                        @o_entry_d      = o_entry_d,
                        @o_carrier_id           = o_carrier_id
                from    orders (serializable)
                where   o_c_id                  = @c_id and
                        o_d_id                  = @d_id and
                        o_w_id                  = @w_id
                order   by o_id asc

-- select order lines for the current order

                select  ol_supply_w_id,
                        ol_i_id,
                        ol_quantity,
                        ol_amount,
                        ol_delivery_d
                from    order_line (repeatableread)
                where   ol_o_id = @o_id and
                        ol_d_id = @d_id and
                        ol_w_id = @w_id

custnotfound:

commit tran o

-- return data to client

select  @c_id,
        @c_last,
        @c_first,
        @c_middle,
        @o_entry_d,
        @o_carrier_id,
        @c_balance,
        @o_id

go
```

APPENDIX B – DATABASE DESIGN

PAYMENT.SQL

```
-- File:    PAYMENT.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21.000
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates payment transaction stored procedure
--
--                  Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_payment" )
                drop procedure tpcc_payment
go

create proc tpcc_payment    @w_id          smallint,
                                                @c_w_id        smallint,
                                                @h_amount      numeric(6,2),
                                                @d_id          tinyint,
                                                @c_d_id        tinyint,
                                                @c_id          int,
                                                @c_last        char(16) = ""


as
declare  @w_street_1    char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   numeric(12,2),
        @c_balance      numeric(12,2),
        @c_discount     numeric(4,4),
        @data           char(500),
        @c_data         char(500),
        @datetime       datetime,
        @w_ytd          numeric(12,2),
        @d_ytd          numeric(12,2),
        @cnt            smallint,
        @val            smallint,
        @screen_data                char(200),
                @d_id_local     tinyint,
                @w_id_local     smallint,
                @c_id_local     int

select @screen_data = ""

begin tran p

-- get payment date

                select          @datetime = getdate()

                if (@c_id = 0)
                begin

-- get customer id and info using last name

                                select          @cnt            = count(*)
                                from            customer (repeatableread)
                                where           c_last          = @c_last and
                                                c_w_id          = @c_w_id and
                                                c_d_id          = @c_d_id


                                select          @val = (@cnt + 1) / 2
                                set             rowcount @val

                                select          @c_id           = c_id
                                from            customer (repeatableread)
                                where           c_last          = @c_last and
                                                c_w_id          = @c_w_id and
                                                c_d_id          = @c_d_id
                                order           by c_last, c_first

                                set             rowcount 0
                end

-- get customer info and update balances

                update          customer
                set             @c_balance      = c_balance     = c_balance - @h_amount,
                                c_payment_cnt                   = c_payment_cnt + 1,
                                c_ytd_payment                   = c_ytd_payment + @h_amount,
                                @c_first        = c_first,
                                @c_middle       = c_middle,
                                @c_last                         = c_last,
                                @c_street_1     = c_street_1,
                                @c_street_2     = c_street_2,
                                @c_city                         = c_city,
```

```
                @c_state    = c_state,
                @c_zip                  = c_zip,
                @c_phone    = c_phone,
                @c_credit   = c_credit,
                @c_credit_lim           = c_credit_lim,
                @c_discount = c_discount,
                @c_since    = c_since,
                @data                   = c_data,
                @c_id_local = c_id
        where   c_id                    = @c_id and
                c_w_id                  = @c_w_id and
                c_d_id                  = @c_d_id

-- if customer has bad credit get some more info

        if (@c_credit = "BC")
        begin

--      compute new info

                select @c_data          = convert(char(5),@c_id) +
                                          convert(char(4),@c_d_id) +
                                          convert(char(5),@c_w_id) +
                                          convert(char(4),@d_id) +
                                          convert(char(5),@w_id) +
                                          convert(char(19),@h_amount) +
                                          substring(@data, 1, 458)

-- update customer info

                update  customer
                set     c_data          = @c_data
                where   c_id            = @c_id and
                        c_w_id          = @c_w_id and
                        c_d_id          = @c_d_id

                select  @screen_data = substring (@c_data,1,200)
        end

-- get district data and update year-to-date

        update  district
        set     d_ytd                   = d_ytd + @h_amount,
                @d_street_1 = d_street_1,
                @d_street_2 = d_street_2,
                @d_city                 = d_city,
                @d_state    = d_state,
                @d_zip                  = d_zip,
                @d_name                 = d_name,
                @d_id_local = d_id
        where   d_w_id                  = @w_id and
                d_id                    = @d_id

-- get warehouse data and update year-to-date

        update  warehouse
        set     w_ytd                   = w_ytd + @h_amount,
                @w_street_1 = w_street_1,
                @w_street_2 = w_street_2,
                @w_city                 = w_city,
                @w_state    = w_state,
                @w_zip                  = w_zip,
                @w_name                 = w_name,
                @w_id_local = w_id
        where   w_id                    = @w_id

-- create history record

        insert into history values (    @c_id_local,
                                        @c_d_id,
                                        @c_w_id,
                                        @d_id_local,
                                        @w_id_local,
                                        @datetime,
                                        @h_amount,
                                        @w_name + "    " + @d_name)
commit tran p

-- return data to client

select  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data

go
```

STOCKLEV.SQL

```
-- File:    STOCKLEV.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21.000
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates stock level transaction stored procedure
--
--                Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_stocklevel" )
                drop procedure tpcc_stocklevel
go

create proc tpcc_stocklevel                     @w_id           smallint,
                                                        @d_id           tinyint,
                                                        @threshhold     smallint
as

declare         @o_id_low int,
                @o_id_high int

select          @o_id_low       = (d_next_o_id - 20),
                @o_id_high      = (d_next_o_id - 1)
from            district
where           d_w_id                          = @w_id and
                d_id                            = @d_id

select          count(distinct(s_i_id))
from            stock, order_line
where           ol_w_id                         =  @w_id and
                ol_d_id                         =  @d_id and
                ol_o_id                           between @o_id_low and
                                                  @o_id_high and
                s_w_id                          =  ol_w_id and
                s_i_id                          =  ol_i_id and
                s_quantity      <  @threshhold

go
```

B.6 Loader Source Code

TPCC.H

```
//              File:                   TPCC.H
//                                              Microsoft TPC-C Kit Ver. 4.21
//                                              Copyright Microsoft, 1996, 1997, 1998, 1999, 2000
//              Purpose:        Header file for TPC-C database loader


// Build number of TPC Benchmark Kit
#define TPCKIT_VER      "4.21"

// General headers
#include <windows.h>
#include <winbase.h>
#include <stdlib.h>
#include <stdio.h>
#include <process.h>
#include <stddef.h>
#include <stdarg.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <sys\types.h>

// ODBC headers
#include <sql.h>
#include <sqlext.h>
#include <odbcss.h>

// General constants
#define MILLI                           1000
#define FALSE                                   0
#define TRUE                                    1
#define UNDEF                                                   -1
#define MINPRINTASCII                           32
#define MAXPRINTASCII                           126

// Default environment constants
#define SERVER                                  ""
#define DATABASE                                "tpcc"
#define USER                                    "sa"
#define PASSWORD                        ""

// Default loader arguments
#define BATCH                           10000
#define DEFLDPACKSIZE                           32768
#define LOADER_RES_FILE                         "logs\\load.out"
#define LOADER_NURAND_C                                 123
#define DEF_STARTING_WAREHOUSE                          1
#define BUILD_INDEX                                             1               // build both data and indexes
#define INDEX_ORDER                                             1               // build indexes before load
#define SCALE_DOWN                                              0  // build a normal scale database
#define INDEX_SCRIPT_PATH               "scripts"

typedef struct
{
   char                                 *server;
   char                                 *database;
   char                                 *user;
```

```
    char                    *password;
        BOOL                    tables_all;             // set if loading all
tables
        BOOL                    table_item;             // set if loading
ITEM table specifically
        BOOL                    table_warehouse;        // set if loading WAREHOUSE,
DISTRICT, and STOCK
        BOOL                    table_customer;         // set if loading
CUSTOMER and HISTORY
        BOOL                    table_orders;           // set if loading NEW-ORDER,
ORDERS, ORDER-LINE
    long                    num_warehouses;
    long                    batch;
    long                    verbose;
        long                    pack_size;
        char                    *loader_res_file;
        char                    *synch_servername;
        long                    case_sensitivity;
        long                    starting_warehouse;
        long                    build_index;
        long                    index_order;
        long                    scale_down;
        char                    *index_script_path;
} TPCCLDR_ARGS;

// String length constants
        #define SERVER_NAME_LEN         20
        #define DATABASE_NAME_LEN       20
        #define USER_NAME_LEN           20
        #define PASSWORD_LEN            20
        #define TABLE_NAME_LEN          20
        #define I_DATA_LEN              50
        #define I_NAME_LEN              24
        #define BRAND_LEN               1
        #define LAST_NAME_LEN           16
        #define W_NAME_LEN              10
        #define ADDRESS_LEN             20
        #define STATE_LEN               2
        #define ZIP_LEN                 9
        #define S_DIST_LEN              24
        #define S_DATA_LEN              50
        #define D_NAME_LEN              10
        #define FIRST_NAME_LEN          16
        #define MIDDLE_NAME_LEN         2
        #define PHONE_LEN               16
        #define CREDIT_LEN              2
        #define C_DATA_LEN              500
        #define H_DATA_LEN              24
        #define DIST_INFO_LEN           24
        #define MAX_OL_NEW_ORDER_ITEMS  15
        #define MAX_OL_ORDER_STATUS_ITEMS 15
        #define STATUS_LEN              25
        #define OL_DIST_INFO_LEN                24
        #define C_SINCE_LEN                             23
        #define H_DATE_LEN                              23
        #define OL_DELIVERY_D_LEN                       23
        #define O_ENTRY_D_LEN           23


// Functions in random.c
void        seed();
long        irand();
double      drand();
void        WUCreate();
short       WURand();
long        RandomNumber(long lower, long upper);

// Functions in getargs.c;
void        GetArgsLoader();
void        GetArgsLoaderUsage();

// Functions in time.c
long        TimeNow();

// Functions in strings.c
void        MakeAddress();
void        LastName();
int         MakeAlphaString();
int         MakeOriginalAlphaString();
int         MakeNumberString();
int         MakeZipNumberString();
void        InitString();
void        InitAddress();
void        PaddString();
```

TPCCLDR.C

```
//          File:               TPCCLDR.C
//                                      Microsoft TPC-C Kit Ver. 4.21
//                                      Copyright Microsoft, 1996, 1997, 1998, 1999, 2000
//          Purpose:    Source file for TPC-C database loader


// Includes
#include "tpcc.h"
#include "search.h"

// Defines
#define MAXITEMS            100000
#define MAXITEMS_SCALE_DOWN                 100
#define CUSTOMERS_PER_DISTRICT  3000
#define CUSTOMERS_SCALE_DOWN    30
#define DISTRICT_PER_WAREHOUSE  10
#define ORDERS_PER_DISTRICT     3000
#define ORDERS_SCALE_DOWN       30
#define MAX_CUSTOMER_THREADS    2
```

APPENDIX B – DATABASE DESIGN

```
#define MAX_ORDER_THREADS      3
#define MAX_MAIN_THREADS       4

// Functions declarations

void HandleErrorDBC (SQLHDBC  hdbc1);

void CheckSQL();
void CheckDataBase();

long NURand();
void LoadItem();
void LoadWarehouse();

void Stock();
void District();

void LoadCustomer();
void CustomerBufInit();
void CustomerBufLoad();
void LoadCustomerTable();
void LoadHistoryTable();

void LoadOrders();
void OrdersBufInit();
void OrdersBufLoad();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void GetPermutation();
void CheckForCommit();
void OpenConnections();
void BuildIndex();
void FormatDate ();

// Shared memory structures

typedef struct
{
   long           ol;
   long           ol_i_id;
   short          ol_supply_w_id;
   short          ol_quantity;
   double          ol_amount;
   char           ol_dist_info[DIST_INFO_LEN+1];
                  char                                    ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCT;

typedef struct
{
   long           o_id;
   short          o_d_id;
   short          o_w_id;
   long           o_c_id;
   short          o_carrier_id;
   short          o_ol_cnt;
   short          o_all_local;
                  ORDER_LINE_STRUCT   o_ol[15];
} ORDERS_STRUCT;

typedef struct
{
   long                                    c_id;
   short                                   c_d_id;
   short                                   c_w_id;
                  char                                    c_first[FIRST_NAME_LEN+1];
   char                                    c_middle[MIDDLE_NAME_LEN+1];
   char                                    c_last[LAST_NAME_LEN+1];
   char                                    c_street_1[ADDRESS_LEN+1];
   char                                    c_street_2[ADDRESS_LEN+1];
   char                                    c_city[ADDRESS_LEN+1];
   char                                    c_state[STATE_LEN+1];
   char                                    c_zip[ZIP_LEN+1];
   char                                    c_phone[PHONE_LEN+1];
   char                                    c_credit[CREDIT_LEN+1];
   double                                  c_credit_lim;
   double                                  c_discount;
//  fix to avoid ODBC float to numeric conversion problem.
//  double                                  c_balance;
                  char                                    c_balance[6];

   double                                  c_ytd_payment;
                  short                                   c_payment_cnt;
                  short                                   c_delivery_cnt;
                  char                                    c_data[C_DATA_LEN+1];
                  double                                  h_amount;
   char                                    h_data[H_DATA_LEN+1];
} CUSTOMER_STRUCT;

typedef struct
{
                  char                                    c_last[LAST_NAME_LEN+1];
                  char                                    c_first[FIRST_NAME_LEN+1];
   long                                    c_id;
} CUSTOMER_SORT_STRUCT;

typedef struct
{
   long           time_start;
} LOADER_TIME_STRUCT;


// Global variables

char           szLastError[300];

HENV           henv;

HDBC           v_hdbc;                                                  // for SQL Server version
verification
```

APPENDIX B – DATABASE DESIGN

```
HDBC        i_hdbc1;                                    // for ITEM table
HDBC        w_hdbc1;                                    // for WAREHOUSE, DISTRICT, STOCK
HDBC        c_hdbc1;                                    // for CUSTOMER
HDBC        c_hdbc2;                                    // for HISTORY
HDBC        o_hdbc1;                                    // for ORDERS
HDBC        o_hdbc2;                                    // for NEW-ORDER

HDBC        o_hdbc3;                                    // for ORDER-LINE

HSTMT       v_hstmt;                                    // for SQL Server version verification
HSTMT       i_hstmt1;
HSTMT       w_hstmt1;
HSTMT       c_hstmt1, c_hstmt2;
HSTMT       o_hstmt1, o_hstmt2, o_hstmt3;

ORDERS_STRUCT   orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT customer_buf[CUSTOMERS_PER_DISTRICT];
long        orders_rows_loaded;
long        new_order_rows_loaded;
long        order_line_rows_loaded;
long        history_rows_loaded;
long        customer_rows_loaded;
long        stock_rows_loaded;
long        district_rows_loaded;
long        item_rows_loaded;
long        warehouse_rows_loaded;
long        main_time_start;
long        main_time_end;
long                    max_items;
long                    customers_per_district;
long                    orders_per_district;
long                    first_new_order;
long                    last_new_order;


TPCCLDR_ARGS    *aptr, args;


//============================================================================
//
// Function name: main
//
//============================================================================

int main(int  argc, char **argv)
{

        DWORD       dwThreadID[MAX_MAIN_THREADS];
        HANDLE      hThread[MAX_MAIN_THREADS];
        FILE        *fLoader;
        char        buffer[255];
        int                                             i;

        for (i=0; i<MAX_MAIN_THREADS; i++)
                hThread[i] = NULL;
```

```
printf("\n*******************************************************");
        printf("\n*                                             *");
        printf("\n*  Microsoft SQL Server                       *");
        printf("\n*                                             *");
        printf("\n*  TPC-C BENCHMARK KIT:  Database loader      *");
        printf("\n*  Version %s                                 *", TPCKIT_VER);
        printf("\n*                                             *");
        printf("\n*******************************************************\n\n");

// process command line arguments

aptr = &args;
GetArgsLoader(argc, argv, aptr);

        // verify database and tables exist before attempting to load

        CheckDataBase();

        printf("Build interface is ODBC.\n");

        if (aptr->build_index == 0)
                printf("Data load only - no index creation.\n");
        else
                printf("Data load and index creation.\n");

        if (aptr->index_order == 0)
                printf("Clustered indexes will be created after bulk load.\n");
        else
                printf("Clustered indexes will be created before bulk load.\n");

        // set database scale values
        if (aptr->scale_down == 1)
        {
                printf("*** Scaled Down Database ***\n");
                max_items = MAXITEMS_SCALE_DOWN;
                customers_per_district = CUSTOMERS_SCALE_DOWN;
                orders_per_district = ORDERS_SCALE_DOWN;
                first_new_order = 0;
                last_new_order  = 30;
        }
        else
        {
                max_items = MAXITEMS;
                customers_per_district = CUSTOMERS_PER_DISTRICT;
                orders_per_district = ORDERS_PER_DISTRICT;
                first_new_order = 2100;
                last_new_order  = 3000;
        }


        // open connections to SQL Server

        OpenConnections();
```

```
        // open file for loader results
fLoader = fopen(aptr->loader_res_file, "w");

if (fLoader == NULL)
        {
                printf("Error, loader result file open failed.");
                exit(-1);
        }

// start loading data

sprintf(buffer,"TPC-C load started for %ld warehouses.\n",aptr->num_warehouses);

        printf("%s",buffer);
        fprintf(fLoader,"%s",buffer);

        main_time_start = (TimeNow() / MILLI);

        // start parallel load threads


        if (aptr->tables_all || aptr->table_item)
        {
                fprintf(fLoader, "\nStarting loader threads for: item\n");

                hThread[0] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadItem,
NULL,
0,
&dwThreadID[0]);

                if (hThread[0] == NULL)
                {
                        printf("Error, failed in creating creating thread = 0.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_warehouse)
        {
                fprintf(fLoader, "Starting loader threads for: warehouse\n");

                hThread[1] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadWarehouse,
NULL,
0,
&dwThreadID[1]);

                if (hThread[1] == NULL)
                {
                        printf("Error, failed in creating creating thread = 1.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_customer)
        {
                fprintf(fLoader, "Starting loader threads for: customer\n");

                hThread[2] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadCustomer,
NULL,
0,
&dwThreadID[2]);

                if (hThread[2] == NULL)
                {
                        printf("Error, failed in creating creating main thread = 2.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_orders)
        {
                fprintf(fLoader, "Starting loader threads for: orders\n");

                hThread[3] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadOrders,
NULL,
0,
&dwThreadID[3]);

                if (hThread[3] == NULL)
                {
                        printf("Error, failed in creating creating main thread = 3.\n");
                        exit(-1);
                }
        }

        // Wait for threads to finish...
        for (i=0; i<MAX_MAIN_THREADS; i++)
        {
                if (hThread[i] != NULL)
                {
                        WaitForSingleObject( hThread[i], INFINITE );
                        CloseHandle(hThread[i]);
                        hThread[i] = NULL;
                }
        }

        main_time_end = (TimeNow() / MILLI);
```

```
    sprintf(buffer,"\nTPC-C load completed successfully in %ld minutes.\n",
                                        (main_time_end - main_time_start)/60);

                printf("%s",buffer);
                fprintf(fLoader, "%s", buffer);

                fclose(fLoader);

                SQLFreeEnv(henv);

    exit(0);

                return 0;
}


//==============================================================================
//
// Function name: LoadItem
//
//==============================================================================

void LoadItem()
{
    long                        i_id;
                long                        i_im_id;
    char                        i_name[I_NAME_LEN+1];
    double                      i_price;
    char                        i_data[I_DATA_LEN+1];
                char                        name[20];
                long                        time_start;
                RETCODE                     rc;
                DBINT                       rcint;
                char                        bcphint[128];

                // Seed with unique number
                seed(1);

                printf("Loading item table...\n");

                // if build index before load
                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                        BuildIndex("idxitmcl");

                InitString(i_name, I_NAME_LEN+1);
                InitString(i_data, I_DATA_LEN+1);

                sprintf(name, "%s..%s", aptr->database, "item");

                rc = bcp_init(i_hdbc1, name, NULL, "logs\\item.err", DB_IN);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                {
                        sprintf(bcphint, "tablock, order (i_id), ROWS_PER_BATCH = 100000");
                        rc = bcp_control(i_hdbc1, BCPHINTS, (void*) bcphint);
                        if (rc != SUCCEED)
                                HandleErrorDBC(i_hdbc1);
                }

                rc = bcp_bind(i_hdbc1, (BYTE *) &i_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

                rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 2);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

                rc = bcp_bind(i_hdbc1, (BYTE *) i_name, 0, I_NAME_LEN, NULL, 0, 0, 3);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

                rc = bcp_bind(i_hdbc1, (BYTE *) &i_price, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 4);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

                rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0, I_DATA_LEN, NULL, 0, 0, 5);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

                time_start = (TimeNow() / MILLI);

                item_rows_loaded = 0;

                for (i_id = 1; i_id <= max_items; i_id++)
                {
                        i_im_id = RandomNumber(1L, 10000L);

                        MakeAlphaString(14, 24, I_NAME_LEN, i_name);

                        i_price = ((float) RandomNumber(100L, 10000L))/100.0;

                        MakeOriginalAlphaString(26, 50, I_DATA_LEN, i_data, 10);

                        rc = bcp_sendrow(i_hdbc1);
                        if (rc != SUCCEED)
                                HandleErrorDBC(i_hdbc1);

                        item_rows_loaded++;
                        CheckForCommit(i_hdbc1, i_hstmt1, item_rows_loaded, "item", &time_start);
                }

                rcint = bcp_done(i_hdbc1);
                if (rcint < 0)
                        HandleErrorDBC(i_hdbc1);

                printf("Finished loading item table.\n");

                SQLFreeStmt(i_hstmt1, SQL_DROP);
```

```
        SQLDisconnect(i_hdbc1);
        SQLFreeConnect(i_hdbc1);

        // if build index after load
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxitmcl");

}


//============================================================================
//
// Function   : LoadWarehouse
//
// Loads WAREHOUSE table and loads Stock and District as Warehouses are created
//
//============================================================================

void LoadWarehouse()
{
   short        w_id;
   char         w_name[W_NAME_LEN+1];
   char         w_street_1[ADDRESS_LEN+1];
   char         w_street_2[ADDRESS_LEN+1];
   char         w_city[ADDRESS_LEN+1];
   char         w_state[STATE_LEN+1];
   char         w_zip[ZIP_LEN+1];
   double       w_tax;
   double       w_ytd;
        char            name[20];
        long            time_start;
        RETCODE         rc;
        DBINT           rcint;
        char            bcphint[128];


        // Seed with unique number
        seed(2);

        printf("Loading warehouse table...\n");

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
                BuildIndex("idxwarcl");

        InitString(w_name, W_NAME_LEN+1);
        InitAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

        sprintf(name, "%s..%s", aptr->database, "warehouse");

        rc = bcp_init(w_hdbc1, name, NULL, "logs\\whouse.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
                sprintf(bcphint, "tablock, order (w_id), ROWS_PER_BATCH = %d", aptr->num_warehouses);
                rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);
}

rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 1);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0, W_NAME_LEN, NULL, 0, 0, 2);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_street_1, 0, ADDRESS_LEN, NULL, 0, 0, 3);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_street_2, 0, ADDRESS_LEN, NULL, 0, 0, 4);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_city, 0, ADDRESS_LEN, NULL, 0, 0, 5);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_state, 0, STATE_LEN, NULL, 0, 0, 6);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_zip, 0, ZIP_LEN, NULL, 0, 0, 7);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 8);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 9);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

time_start = (TimeNow() / MILLI);

warehouse_rows_loaded = 0;

for (w_id = (short)aptr->starting_warehouse; w_id <= aptr->num_warehouses; w_id++)
{
        MakeAlphaString(6,10, W_NAME_LEN, w_name);

        MakeAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

        w_tax = ((float) RandomNumber(0L,2000L))/10000.00;
```

```
                w_ytd = 300000.00;

                rc = bcp_sendrow(w_hdbc1);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                warehouse_rows_loaded++;
                CheckForCommit(w_hdbc1, i_hstmt1, warehouse_rows_loaded, "warehouse", &time_start);
        }

        rcint = bcp_done(w_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(w_hdbc1);

        printf("Finished loading warehouse table.\n");

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxwarcl");

        stock_rows_loaded = 0;
        district_rows_loaded = 0;

        District();
        Stock();

}


//============================================================================
//
// Function   : District
//
//============================================================================

void District()
{
   short        d_id;
   short        d_w_id;
   char         d_name[D_NAME_LEN+1];
   char         d_street_1[ADDRESS_LEN+1];
   char         d_street_2[ADDRESS_LEN+1];
   char         d_city[ADDRESS_LEN+1];
   char         d_state[STATE_LEN+1];
   char         d_zip[ZIP_LEN+1];
   double       d_tax;
   double       d_ytd;
                char            name[20];
   long         d_next_o_id;
                long            time_start;
                int                             w_id;
                RETCODE         rc;
                DBINT           rcint;
                char            bcphint[128];

                // Seed with unique number
                seed(4);

                printf("Loading district table...\n");

                // build index before load
                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                        BuildIndex("idxdiscl");

                InitString(d_name, D_NAME_LEN+1);
                InitAddress(d_street_1, d_street_2, d_city, d_state, d_zip);
                sprintf(name, "%s..%s", aptr->database, "district");

                rc = bcp_init(w_hdbc1, name, NULL, "logs\\district.err", DB_IN);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                {
                        sprintf(bcphint, "tablock, order (d_w_id, d_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses
* 10));
                        rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
                        if (rc != SUCCEED)
                                HandleErrorDBC(w_hdbc1);
                }

                rc = bcp_bind(w_hdbc1, (BYTE *) &d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 1);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) &d_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) d_name, 0, D_NAME_LEN, NULL, 0, 0, 3);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) d_street_1, 0, ADDRESS_LEN, NULL, 0, 0, 4);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) d_street_2, 0, ADDRESS_LEN, NULL, 0, 0, 5);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) d_city, 0, ADDRESS_LEN, NULL, 0, 0, 6);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) d_state, 0, STATE_LEN, NULL, 0, 0, 7);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) d_zip, 0, ZIP_LEN, NULL, 0, 0, 8);
```

```
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &d_tax, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 9);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &d_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 10);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &d_next_o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 11);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        d_ytd  = 30000.0;

        d_next_o_id = orders_per_district+1;

        time_start = (TimeNow() / MILLI);

        for (w_id = aptr->starting_warehouse; w_id <= aptr->num_warehouses; w_id++)
        {
                d_w_id = w_id;

                for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
                {
                        MakeAlphaString(6,10,D_NAME_LEN, d_name);

                        MakeAddress(d_street_1, d_street_2, d_city, d_state, d_zip);

                        d_tax = ((float) RandomNumber(0L,2000L))/10000.00;

                        rc = bcp_sendrow(w_hdbc1);
                        if (rc != SUCCEED)
                                HandleErrorDBC(w_hdbc1);

                        district_rows_loaded++;
                        CheckForCommit(w_hdbc1, w_hstmt1, district_rows_loaded, "district", &time_start);
                }
  }

        rcint = bcp_done(w_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(w_hdbc1);

        printf("Finished loading district table.\n");

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxdiscl");

  return;
}
```

```
//==============================================================================
//
// Function   : Stock
//
//==============================================================================

void Stock()
{
   long         s_i_id;
   short        s_w_id;
   short        s_quantity;
   char         s_dist_01[S_DIST_LEN+1];
   char         s_dist_02[S_DIST_LEN+1];
   char         s_dist_03[S_DIST_LEN+1];
   char         s_dist_04[S_DIST_LEN+1];
   char         s_dist_05[S_DIST_LEN+1];
   char         s_dist_06[S_DIST_LEN+1];
   char         s_dist_07[S_DIST_LEN+1];
   char         s_dist_08[S_DIST_LEN+1];
   char         s_dist_09[S_DIST_LEN+1];
   char         s_dist_10[S_DIST_LEN+1];
   long         s_ytd;
   short        s_order_cnt;
   short        s_remote_cnt;
   char         s_data[S_DATA_LEN+1];
   short        len;
                char            name[20];
                long            time_start;
                RETCODE         rc;
                DBINT           rcint;
                char            bcphint[128];

                // Seed with unique number
                seed(3);

                // if build index before load...
                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                        BuildIndex("idxstkcl");

                sprintf(name, "%s..%s", aptr->database, "stock");

                rc = bcp_init(w_hdbc1, name, NULL, "logs\\stock.err", DB_IN);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                {
                        sprintf(bcphint, "tablock, order (s_i_id, s_w_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 100000));
                        rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
                        if (rc != SUCCEED)
                                HandleErrorDBC(w_hdbc1);
                }
```

```
rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_quantity, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01, 0, S_DIST_LEN, NULL, 0, 0, 4);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02, 0, S_DIST_LEN, NULL, 0, 0, 5);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03, 0, S_DIST_LEN, NULL, 0, 0, 6);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04, 0, S_DIST_LEN, NULL, 0, 0, 7);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05, 0, S_DIST_LEN, NULL, 0, 0, 8);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06, 0, S_DIST_LEN, NULL, 0, 0, 9);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07, 0, S_DIST_LEN, NULL, 0, 0, 10);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0, S_DIST_LEN, NULL, 0, 0, 11);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09, 0, S_DIST_LEN, NULL, 0, 0, 12);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0, S_DIST_LEN, NULL, 0, 0, 13);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 14);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_order_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 15);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_remote_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 16);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0, S_DATA_LEN, NULL, 0, 0, 17);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

s_ytd = s_order_cnt = s_remote_cnt = 0;

time_start = (TimeNow() / MILLI);

printf("...Loading stock table\n");

for (s_i_id=1; s_i_id <= max_items; s_i_id++)
{

        for (s_w_id = (short)aptr->starting_warehouse; s_w_id <= aptr->num_warehouses; s_w_id++)
        {

                s_quantity = (short)RandomNumber(10L,100L);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_01);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_02);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_03);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_04);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_05);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_06);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_07);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_08);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_09);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_10);

                len = MakeOriginalAlphaString(26,50, S_DATA_LEN, s_data,10);

                rc = bcp_sendrow(w_hdbc1);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                stock_rows_loaded++;
                CheckForCommit(w_hdbc1, w_hstmt1, stock_rows_loaded, "stock", &time_start);

        }
}

rcint = bcp_done(w_hdbc1);
if (rcint < 0)
        HandleErrorDBC(w_hdbc1);

printf("Finished loading stock table.\n");
```

```
		SQLFreeStmt(w_hstmt1, SQL_DROP);
		SQLDisconnect(w_hdbc1);
		SQLFreeConnect(w_hdbc1);

		// if build index after load...
		if ((aptr->build_index == 1) && (aptr->index_order == 0))
			BuildIndex("idxstkcl");

		return;
}


//=============================================================================
//
// Function   : LoadCustomer
//
//=============================================================================

void LoadCustomer()
{
		LOADER_TIME_STRUCT	customer_time_start;
		LOADER_TIME_STRUCT	history_time_start;
		short			w_id;
  short				d_id;
		DWORD				dwThreadID[MAX_CUSTOMER_THREADS];
		HANDLE				hThread[MAX_CUSTOMER_THREADS];
		char				name[20];
		RETCODE					rc;
		DBINT					rcint;
		char					bcphint[128];
		char					cmd[256];
		// SQLRETURN				rc_1;
		// SQLSMALLINT				recnum, MsgLen;
		// SQLCHAR				SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
		// SQLINTEGER				NativeError;

		// Seed with unique number
		seed(5);

		printf("Loading customer and history tables...\n");

		// if build index before load...
		if ((aptr->build_index == 1) && (aptr->index_order == 1))
			BuildIndex("idxcuscl");

		// Initialize bulk copy
		sprintf(name, "%s..%s", aptr->database, "customer");

		rc = bcp_init(c_hdbc1, name, NULL, "logs\\customer.err", DB_IN);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc1);

		if ((aptr->build_index == 1) && (aptr->index_order == 1))
		{
			sprintf(bcphint, "tablock, order (c_w_id, c_d_id, c_id), ROWS_PER_BATCH = %u", (aptr-
>num_warehouses * 30000));
			rc = bcp_control(c_hdbc1, BCPHINTS, (void*) bcphint);
			if (rc != SUCCEED)
				HandleErrorDBC(c_hdbc1);
		}

		sprintf(name, "%s..%s", aptr->database, "history");

		rc = bcp_init(c_hdbc2, name, NULL, "logs\\history.err", DB_IN);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc2);

		sprintf(bcphint, "tablock");
		rc = bcp_control(c_hdbc2, BCPHINTS, (void*) bcphint);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc2);

		customer_rows_loaded	= 0;
		history_rows_loaded	= 0;

		CustomerBufInit();

		customer_time_start.time_start = (TimeNow() / MILLI);
		history_time_start.time_start = (TimeNow() / MILLI);

		for (w_id = (short)aptr->starting_warehouse; w_id <= aptr->num_warehouses; w_id++)
		{
			for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
			{
				CustomerBufLoad(d_id, w_id);

				// Start parallel loading threads here...

				// Start customer table thread

				printf("...Loading customer table for: d_id = %d, w_id = %d\n", d_id, w_id);

				hThread[0] = CreateThread(NULL,
0,

(LPTHREAD_START_ROUTINE) LoadCustomerTable,

&customer_time_start,
0,

&dwThreadID[0]);

				if (hThread[0] == NULL)
				{
					printf("Error, failed in creating creating thread = 0.\n");
					exit(-1);
				}
```

```
                        // Start History table thread

                        printf("...Loading history table for: d_id = %d, w_id = %d\n", d_id, w_id);

                        hThread[1] = CreateThread(NULL,
                                                                                                0,
(LPTHREAD_START_ROUTINE) LoadHistoryTable,

&history_time_start,
                                                                                                0,

&dwThreadID[1]);

                        if (hThread[1] == NULL)
                        {
                                printf("Error, failed in creating creating thread = 1.\n");
                                exit(-1);
                        }

                        WaitForSingleObject( hThread[0], INFINITE );
                        WaitForSingleObject( hThread[1], INFINITE );

                        if (CloseHandle(hThread[0]) == FALSE)
                        {
                                printf("Error, failed in closing customer thread handle with errno: %d\n",
GetLastError());
                        }

                        if (CloseHandle(hThread[1]) == FALSE)
                        {
                                printf("Error, failed in closing history thread handle with errno: %d\n",
GetLastError());
                        }


                }

        }

        // flush the bulk connection
        rcint = bcp_done(c_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(c_hdbc1);

        rcint = bcp_done(c_hdbc2);
        if (rcint < 0)
                HandleErrorDBC(c_hdbc2);

        printf("Finished loading customer table.\n");

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxcuscl");

        // build non-clustered index
        if (aptr->build_index == 1)
                BuildIndex("idxcusnc");

        // Output the NURAND used for the loader into C_FIRST for C_ID = 1,
        // C_W_ID = 1, and C_D_ID = 1
        sprintf(cmd, "isql -S%s -U%s -P%s -d%s -e -Q\"update customer set c_first = 'C_LOAD = %d' where c_id = 1 and
c_w_id = 1 and c_d_id = 1\" > logs\\nurand_load.log",
                                                        aptr->server,
                                                        aptr->user,
                                                        aptr->password,
                                                        aptr->database,
                                                        LOADER_NURAND_C);

        system(cmd);

        SQLFreeStmt(c_hstmt1, SQL_DROP);
        SQLDisconnect(c_hdbc1);
        SQLFreeConnect(c_hdbc1);

        SQLFreeStmt(c_hstmt2, SQL_DROP);
        SQLDisconnect(c_hdbc2);
        SQLFreeConnect(c_hdbc2);


    return;

}


//====================================================================
//
// Function   : CustomerBufInit
//
//====================================================================

void CustomerBufInit()
{
        int     i;

        for (i=0;i<customers_per_district;i++)
        {
                customer_buf[i].c_id = 0;
                customer_buf[i].c_d_id = 0;
                customer_buf[i].c_w_id = 0;

                strcpy(customer_buf[i].c_first,"");
                strcpy(customer_buf[i].c_middle,"");
                strcpy(customer_buf[i].c_last,"");
                strcpy(customer_buf[i].c_street_1,"");
                strcpy(customer_buf[i].c_street_2,"");
                strcpy(customer_buf[i].c_city,"");
                strcpy(customer_buf[i].c_state,"");
                strcpy(customer_buf[i].c_zip,"");
                strcpy(customer_buf[i].c_phone,"");
```

```
                strcpy(customer_buf[i].c_credit,"");

                customer_buf[i].c_credit_lim = 0;
                customer_buf[i].c_discount = (float) 0;

                //  fix to avoid ODBC float to numeric conversion problem.
                //              customer_buf[i].c_balance = 0;
                strcpy(customer_buf[i].c_balance,"");

                customer_buf[i].c_ytd_payment = 0;
                customer_buf[i].c_payment_cnt = 0;
                customer_buf[i].c_delivery_cnt = 0;

                strcpy(customer_buf[i].c_data,"");

                customer_buf[i].h_amount = 0;

                strcpy(customer_buf[i].h_data,"");

        }


}


//================================================================================
//
// Function   : CustomerBufLoad
//
// Fills shared buffer for HISTORY and CUSTOMER
//================================================================================

void CustomerBufLoad(int d_id, int w_id)
{
        long                                                    i;
   CUSTOMER_SORT_STRUCT    c[CUSTOMERS_PER_DISTRICT];


        for (i=0;i<customers_per_district;i++)
        {
                if (i < 1000)
                        LastName(i, c[i].c_last);
                else
                        LastName(NURand(255,0,999,LOADER_NURAND_C), c[i].c_last);

                MakeAlphaString(8,16,FIRST_NAME_LEN, c[i].c_first);

                c[i].c_id = i+1;

        }

        printf("...Loading customer buffer for: d_id = %d, w_id = %d\n",
                        d_id, w_id);

        for (i=0;i<customers_per_district;i++)
        {

                customer_buf[i].c_d_id = d_id;
                customer_buf[i].c_w_id = w_id;
                customer_buf[i].h_amount = 10.0;

                customer_buf[i].c_ytd_payment  = 10.0;

                customer_buf[i].c_payment_cnt  = 1;
                customer_buf[i].c_delivery_cnt = 0;

                // Generate CUSTOMER and HISTORY data

                customer_buf[i].c_id = c[i].c_id;

                strcpy(customer_buf[i].c_first, c[i].c_first);
                strcpy(customer_buf[i].c_last, c[i].c_last);

                customer_buf[i].c_middle[0] = 'O';
                customer_buf[i].c_middle[1] = 'E';

                MakeAddress(customer_buf[i].c_street_1,
                                        customer_buf[i].c_street_2,
                                        customer_buf[i].c_city,
                                        customer_buf[i].c_state,
                                        customer_buf[i].c_zip);

                MakeNumberString(16, 16, PHONE_LEN, customer_buf[i].c_phone);

                if (RandomNumber(1L, 100L) > 10)
                        customer_buf[i].c_credit[0] = 'G';
                else
                        customer_buf[i].c_credit[0] = 'B';
                customer_buf[i].c_credit[1] = 'C';

                customer_buf[i].c_credit_lim = 50000.0;
                customer_buf[i].c_discount = ((float) RandomNumber(0L, 5000L)) / 10000.0;


                //  fix to avoid ODBC float to numeric conversion problem.
                //  customer_buf[i].c_balance = -10.0;
                strcpy(customer_buf[i].c_balance,"-10.0");

                MakeAlphaString(300, 500, C_DATA_LEN, customer_buf[i].c_data);

                // Generate HISTORY data
                MakeAlphaString(12, 24, H_DATA_LEN, customer_buf[i].h_data);

        }
}

//================================================================================
//
// Function   : LoadCustomerTable
//
```

```
//====================================================================

void LoadCustomerTable(LOADER_TIME_STRUCT *customer_time_start)
{
                int     i;
    long    c_id;
    short   c_d_id;
    short   c_w_id;
    char    c_first[FIRST_NAME_LEN+1];
    char    c_middle[MIDDLE_NAME_LEN+1];
    char    c_last[LAST_NAME_LEN+1];
    char    c_street_1[ADDRESS_LEN+1];
    char    c_street_2[ADDRESS_LEN+1];
    char    c_city[ADDRESS_LEN+1];
    char    c_state[STATE_LEN+1];
    char    c_zip[ZIP_LEN+1];
    char    c_phone[PHONE_LEN+1];
    char    c_credit[CREDIT_LEN+1];
    double  c_credit_lim;
    double  c_discount;

                //  fix to avoid ODBC float to numeric conversion problem.
                //  double                  c_balance;
                char                    c_balance[6];

    double  c_ytd_payment;
    short   c_payment_cnt;
    short   c_delivery_cnt;
    char    c_data[C_DATA_LEN+1];
                char                    c_since[C_SINCE_LEN+1];
                RETCODE                 rc;

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

                rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0, FIRST_NAME_LEN, NULL, 0, 0, 4);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0, MIDDLE_NAME_LEN,NULL, 0, 0, 5);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN, NULL, 0, 0, 6);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN, NULL, 0, 0, 7);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN, NULL,0,0, 8);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL, 0, 0, 9);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0, 0, 10);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0, 11);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL, 0, 0, 12);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

                rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0, C_SINCE_LEN, NULL, 0, SQLCHARACTER, 13);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL, 0, 0, 14);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 15);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 16);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);


                //  fix to avoid ODBC float to numeric conversion problem.
//  rc = bcp_bind(c_hdbc1, (BYTE *) &c_balance, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 17);
                //  if (rc != SUCCEED)
                //              HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0, SQLCHARACTER, 17);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);


rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 18);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);
```

```
   rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 19);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

   rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt,0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 20);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

   rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0, 500, NULL, 0, 0, 21);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

   for (i = 0; i < customers_per_district; i++)
   {
                        c_id = customer_buf[i].c_id;
                        c_d_id = customer_buf[i].c_d_id;
                        c_w_id = customer_buf[i].c_w_id;

                        strcpy(c_first, customer_buf[i].c_first);
                        strcpy(c_middle, customer_buf[i].c_middle);
                        strcpy(c_last, customer_buf[i].c_last);
                        strcpy(c_street_1, customer_buf[i].c_street_1);
                        strcpy(c_street_2, customer_buf[i].c_street_2);
                        strcpy(c_city, customer_buf[i].c_city);
                        strcpy(c_state, customer_buf[i].c_state);
                        strcpy(c_zip, customer_buf[i].c_zip);
                        strcpy(c_phone, customer_buf[i].c_phone);
                        strcpy(c_credit, customer_buf[i].c_credit);

                        FormatDate(&c_since);

                        c_credit_lim = customer_buf[i].c_credit_lim;
                        c_discount = customer_buf[i].c_discount;

                        //  fix to avoid ODBC float to numeric conversion problem.
                        //  c_balance = customer_buf[i].c_balance;
                        strcpy(c_balance, customer_buf[i].c_balance);

                        c_ytd_payment = customer_buf[i].c_ytd_payment;
                        c_payment_cnt = customer_buf[i].c_payment_cnt;
                        c_delivery_cnt = customer_buf[i].c_delivery_cnt;

                        strcpy(c_data, customer_buf[i].c_data);

                        // Send data to server
                        rc = bcp_sendrow(c_hdbc1);
                        if (rc != SUCCEED)
                                HandleErrorDBC(c_hdbc1);

                        customer_rows_loaded++;
                        CheckForCommit(c_hdbc1, c_hstmt1, customer_rows_loaded, "customer", &customer_time_start-
>time_start);
   }

}
```

```
//===========================================================================
//
// Function   : LoadHistoryTable
//
//===========================================================================

void LoadHistoryTable(LOADER_TIME_STRUCT *history_time_start)
{
                int     i;
   long     c_id;
   short    c_d_id;
   short    c_w_id;
                double          h_amount;
   char     h_data[H_DATA_LEN+1];
                char                    h_date[H_DATE_LEN+1];
                RETCODE                 rc;

   rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

   rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

   rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

   rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 4);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

   rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 5);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

                rc = bcp_bind(c_hdbc2, (BYTE *) &h_date, 0, H_DATE_LEN, NULL, 0, SQLCHARACTER, 6);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

   rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 7);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

   rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0, H_DATA_LEN, NULL, 0, 0, 8);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

   for (i = 0; i < customers_per_district; i++)
   {
                        c_id = customer_buf[i].c_id;
                        c_d_id = customer_buf[i].c_d_id;
```

```
                        c_w_id = customer_buf[i].c_w_id;
                        h_amount = customer_buf[i].h_amount;
                        strcpy(h_data, customer_buf[i].h_data);

                        FormatDate(&h_date);

                        // send to server
                        rc = bcp_sendrow(c_hdbc2);
                        if (rc != SUCCEED)
                                HandleErrorDBC(c_hdbc2);

                        history_rows_loaded++;
                        CheckForCommit(c_hdbc2, c_hstmt2, history_rows_loaded, "history", &history_time_start->time_start);
    }

}


//=============================================================-================
//
// Function   : LoadOrders
//
//=============================================================================

void LoadOrders()
{
        LOADER_TIME_STRUCT      orders_time_start;
        LOADER_TIME_STRUCT      new_order_time_start;
        LOADER_TIME_STRUCT      order_line_time_start;
        short                   w_id;
    short               d_id;
        DWORD                           dwThreadID[MAX_ORDER_THREADS];
        HANDLE                          hThread[MAX_ORDER_THREADS];
        char                    name[20];
        RETCODE                                                 rc;
        char                                                    bcphint[128];

        // seed with unique number
        seed(6);

        printf("Loading orders...\n");

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                        BuildIndex("idxordcl");
                        BuildIndex("idxnodcl");
                        BuildIndex("idxodlcl");
        }

        // initialize bulk copy
        sprintf(name, "%s..%s", aptr->database, "orders");

        rc = bcp_init(o_hdbc1, name, NULL, "logs\\orders.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (o_w_id, o_d_id, o_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
                rc = bcp_control(o_hdbc1, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);
        }

        sprintf(name, "%s..%s", aptr->database, "new_order");

        rc = bcp_init(o_hdbc2, name, NULL, "logs\\neword.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (no_w_id, no_d_id, no_o_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 9000));
                rc = bcp_control(o_hdbc2, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc2);
        }

        sprintf(name, "%s..%s", aptr->database, "order_line");

        rc = bcp_init(o_hdbc3, name, NULL, "logs\\ordline.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (ol_w_id, ol_d_id, ol_o_id, ol_number), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 300000));
                rc = bcp_control(o_hdbc3, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
        }

        orders_rows_loaded      = 0;
        new_order_rows_loaded   = 0;
        order_line_rows_loaded  = 0;

        OrdersBufInit();

        orders_time_start.time_start = (TimeNow() / MILLI);
        new_order_time_start.time_start = (TimeNow() / MILLI);
        order_line_time_start.time_start = (TimeNow() / MILLI);

        for (w_id = (short)aptr->starting_warehouse; w_id <= aptr->num_warehouses; w_id++)
        {
                for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
```

```
                {
                        OrdersBufLoad(d_id, w_id);

                        // start parallel loading threads here...

                        // start Orders table thread

                        printf("...Loading Order Table for: d_id = %d, w_id = %d\n", d_id, w_id);

                        hThread[0] = CreateThread(NULL,
                                                                                0,
(LPTHREAD_START_ROUTINE) LoadOrdersTable,
&orders_time_start,
                                                                                0,
&dwThreadID[0]);
                        if (hThread[0] == NULL)
                        {
                                printf("Error, failed in creating creating thread = 0.\n");
                                exit(-1);
                        }

                        // start NewOrder table thread

                        printf("...Loading New-Order Table for: d_id = %d, w_id = %d\n", d_id, w_id);

                        hThread[1] = CreateThread(NULL,
                                                                                0,
(LPTHREAD_START_ROUTINE) LoadNewOrderTable,
&new_order_time_start,
                                                                                0,
&dwThreadID[1]);
                        if (hThread[1] == NULL)
                        {
                                printf("Error, failed in creating creating thread = 1.\n");
                                exit(-1);
                        }

                        // start Order-Line table thread

                        printf("...Loading Order-Line Table for: d_id = %d, w_id = %d\n", d_id, w_id);

                        hThread[2] = CreateThread(NULL,
                                                                                0,
(LPTHREAD_START_ROUTINE) LoadOrderLineTable,
&order_line_time_start,
                                                                                0,
&dwThreadID[2]);
                        if (hThread[2] == NULL)
                        {
                                printf("Error, failed in creating creating thread = 2.\n");
                                exit(-1);
                        }

                        WaitForSingleObject( hThread[0], INFINITE );
                        WaitForSingleObject( hThread[1], INFINITE );
                        WaitForSingleObject( hThread[2], INFINITE );

                        if (CloseHandle(hThread[0]) == FALSE)
                        {
                                printf("Error, failed in closing Orders thread handle with errno: %d\n",
GetLastError());
                        }

                        if (CloseHandle(hThread[1]) == FALSE)
                        {
                                printf("Error, failed in closing NewOrder thread handle with errno: %d\n",
GetLastError());
                        }

                        if (CloseHandle(hThread[2]) == FALSE)
                        {
                                printf("Error, failed in closing OrderLine thread handle with errno: %d\n",
GetLastError());
                        }

                }
        }
        printf("Finished loading orders.\n");

    return;
}


//========================================================================
//
// Function   : OrdersBufInit
//
// Clears shared buffer for ORDERS, NEWORDER, and ORDERLINE
//
//========================================================================

void OrdersBufInit()
{
        int     i;
```

```
        int                     j;

        for (i=0;i<orders_per_district;i++)
        {
                orders_buf[i].o_id = 0;
                orders_buf[i].o_d_id = 0;
                orders_buf[i].o_w_id = 0;
                orders_buf[i].o_c_id = 0;
                orders_buf[i].o_carrier_id = 0;
                orders_buf[i].o_ol_cnt = 0;
                orders_buf[i].o_all_local = 0;

                for (j=0;j<=14;j++)
                {
                        orders_buf[i].o_ol[j].ol = 0;
                        orders_buf[i].o_ol[j].ol_i_id = 0;
                        orders_buf[i].o_ol[j].ol_supply_w_id = 0;
                        orders_buf[i].o_ol[j].ol_quantity = 0;
                        orders_buf[i].o_ol[j].ol_amount =  0;
                        strcpy(orders_buf[i].o_ol[j].ol_dist_info,"");
                }
        }


}


//============================================================================
//
// Function   : OrdersBufLoad
//
// Fills shared buffer for ORDERS, NEWORDER, and ORDERLINE
//
//============================================================================

void OrdersBufLoad(int d_id, int w_id)
{

        int     cust[ORDERS_PER_DISTRICT+1];
        long    o_id;
        short   ol;

        printf("...Loading Order Buffer for: d_id = %d, w_id = %d\n",
                        d_id, w_id);

        GetPermutation(cust, orders_per_district);

        for (o_id=0;o_id<orders_per_district;o_id++)
        {

                // Generate ORDER and NEW-ORDER data

                orders_buf[o_id].o_d_id = d_id;
                orders_buf[o_id].o_w_id = w_id;
                orders_buf[o_id].o_id = o_id+1;
                orders_buf[o_id].o_c_id = cust[o_id+1];
                orders_buf[o_id].o_ol_cnt = (short)RandomNumber(5L, 15L);

                if (o_id < first_new_order)
                {
                        orders_buf[o_id].o_carrier_id = (short)RandomNumber(1L, 10L);
                        orders_buf[o_id].o_all_local  = 1;
                }
                else
                {
                        orders_buf[o_id].o_carrier_id = 0;
                        orders_buf[o_id].o_all_local  = 1;
                }

                for (ol=0; ol<orders_buf[o_id].o_ol_cnt; ol++)
                {

                        orders_buf[o_id].o_ol[ol].ol = ol+1;
                        orders_buf[o_id].o_ol[ol].ol_i_id = RandomNumber(1L, max_items);
                        orders_buf[o_id].o_ol[ol].ol_supply_w_id = w_id;
                        orders_buf[o_id].o_ol[ol].ol_quantity = 5;
                        MakeAlphaString(24, 24, OL_DIST_INFO_LEN, &orders_buf[o_id].o_ol[ol].ol_dist_info);

                        // Generate ORDER-LINE data
                        if (o_id < first_new_order)
                        {
                                orders_buf[o_id].o_ol[ol].ol_amount = 0;
                                // Added to insure ol_delivery_d set properly during load

                                FormatDate(&orders_buf[o_id].o_ol[ol].ol_delivery_d);

                        }
                        else
                        {
                                orders_buf[o_id].o_ol[ol].ol_amount = RandomNumber(1,999999)/100.0;
                                // Added to insure ol_delivery_d set properly during load


                                // odbc datetime format
                                strcpy(orders_buf[o_id].o_ol[ol].ol_delivery_d,"1899-12-31 00:00:00.000");

                        }
                }
        }
}


//============================================================================
//
// Function   : LoadOrdersTable
//
//============================================================================

void LoadOrdersTable(LOADER_TIME_STRUCT *orders_time_start)
{
```

```
        int     i;
long    o_id;
        short   o_d_id;
        short   o_w_id;
long    o_c_id;
short   o_carrier_id;
short   o_ol_cnt;
short   o_all_local;
        char                    o_entry_d[O_ENTRY_D_LEN+1];
        RETCODE                 rc;
        DBINT                   rcint;

        // bind ORDER data
rc = bcp_bind(o_hdbc1, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

rc = bcp_bind(o_hdbc1, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

        rc = bcp_bind(o_hdbc1, (BYTE *) &o_entry_d, 0, O_ENTRY_D_LEN, NULL, 0, SQLCHARACTER, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 6);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 7);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 8);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

for (i = 0; i < orders_per_district; i++)
{
                o_id        = orders_buf[i].o_id;
                o_d_id      = orders_buf[i].o_d_id;
                o_w_id      = orders_buf[i].o_w_id;
                o_c_id      = orders_buf[i].o_c_id;
                o_carrier_id = orders_buf[i].o_carrier_id;
                o_ol_cnt    = orders_buf[i].o_ol_cnt;
                o_all_local  = orders_buf[i].o_all_local;
                FormatDate(&o_entry_d);

                // send data to server
                rc = bcp_sendrow(o_hdbc1);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);

                orders_rows_loaded++;
                CheckForCommit(o_hdbc1, o_hstmt1, orders_rows_loaded, "orders", &orders_time_start-
>time_start);
   }

        // rcint = bcp_batch(o_hdbc1);
        // if (rcint < 0)
        //      HandleErrorDBC(o_hdbc1);

        if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
        {
                rcint = bcp_done(o_hdbc1);
                if (rcint < 0)
                        HandleErrorDBC(o_hdbc1);

                SQLFreeStmt(o_hstmt1, SQL_DROP);
                SQLDisconnect(o_hdbc1);
                SQLFreeConnect(o_hdbc1);

                // if build index after load...
                if ((aptr->build_index == 1) && (aptr->index_order == 0))
                        BuildIndex("idxordcl");

                // build non-clustered index
                if (aptr->build_index == 1)
                        BuildIndex("idxordnc");
        }

}


//=============================================================================
//
// Function   : LoadNewOrderTable
//
//=============================================================================

void LoadNewOrderTable(LOADER_TIME_STRUCT *new_order_time_start)
{
   int     i;
   long    o_id;
   short   o_d_id;
   short   o_w_id;
        RETCODE                 rc;
        DBINT                   rcint;

        // Bind NEW-ORDER data
```

```
    rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
                if (rc != SUCCEED)
                            HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
                if (rc != SUCCEED)
                            HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
                if (rc != SUCCEED)
                            HandleErrorDBC(o_hdbc2);

                for (i = first_new_order; i < last_new_order; i++)
                {
                            o_id    = orders_buf[i].o_id;
                            o_d_id  = orders_buf[i].o_d_id;
                            o_w_id  = orders_buf[i].o_w_id;

                            rc = bcp_sendrow(o_hdbc2);
                            if (rc != SUCCEED)
                                        HandleErrorDBC(o_hdbc2);

                            new_order_rows_loaded++;
                            CheckForCommit(o_hdbc2, o_hstmt2, new_order_rows_loaded, "new_order",
&new_order_time_start->time_start);
                }

                // rcint = bcp_batch(o_hdbc2);
                // if (rcint < 0)
                //          HandleErrorDBC(o_hdbc2);

                if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
                {
                            rcint = bcp_done(o_hdbc2);
                            if (rcint < 0)
                                        HandleErrorDBC(o_hdbc2);

                            SQLFreeStmt(o_hstmt2, SQL_DROP);
                            SQLDisconnect(o_hdbc2);
                            SQLFreeConnect(o_hdbc2);

                            // if build index after load...
                            if ((aptr->build_index == 1) && (aptr->index_order == 0))
                                        BuildIndex("idxnodcl");

                }

}


//==============================================================================
//
// Function   : LoadOrderLineTable
//
//==============================================================================

void LoadOrderLineTable(LOADER_TIME_STRUCT *order_line_time_start)
{
                int       i,j;
    long      o_id;
                short           o_d_id;
                short           o_w_id;
    long      ol;
                long            ol_i_id;
    short     ol_supply_w_id;
    short     ol_quantity;
    double    ol_amount;
    char      ol_dist_info[DIST_INFO_LEN+1];
                char                        ol_delivery_d[OL_DELIVERY_D_LEN+1];
                RETCODE                     rc;
                DBINT                       rcint;

                // bind ORDER-LINE data
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
                if (rc != SUCCEED)
                            HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
                if (rc != SUCCEED)
                            HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
                if (rc != SUCCEED)
                            HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 4);
                if (rc != SUCCEED)
                            HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 5);
                if (rc != SUCCEED)
                            HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 6);
                if (rc != SUCCEED)
                            HandleErrorDBC(o_hdbc3);

                rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0, OL_DELIVERY_D_LEN, NULL, 0, SQLCHARACTER, 7);
                if (rc != SUCCEED)
                            HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 8);
                if (rc != SUCCEED)
                            HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 9);
                if (rc != SUCCEED)
                            HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0, DIST_INFO_LEN, NULL, 0, 0, 10);
```

```
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

        for (i = 0; i < orders_per_district; i++)
  {
                o_id    = orders_buf[i].o_id;
                o_d_id  = orders_buf[i].o_d_id;
                o_w_id  = orders_buf[i].o_w_id;

                for (j=0; j < orders_buf[i].o_ol_cnt; j++)
                {
                        ol          = orders_buf[i].o_ol[j].ol;
                        ol_i_id     = orders_buf[i].o_ol[j].ol_i_id;
                        ol_supply_w_id = orders_buf[i].o_ol[j].ol_supply_w_id;
                        ol_quantity   = orders_buf[i].o_ol[j].ol_quantity;
                        ol_amount     = orders_buf[i].o_ol[j].ol_amount;
                        strcpy(ol_delivery_d,orders_buf[i].o_ol[j].ol_delivery_d);

                        strcpy(ol_dist_info,orders_buf[i].o_ol[j].ol_dist_info);

                        rc = bcp_sendrow(o_hdbc3);
                        if (rc != SUCCEED)
                                HandleErrorDBC(o_hdbc3);

                        order_line_rows_loaded++;
                        CheckForCommit(o_hdbc3, o_hstmt3, order_line_rows_loaded, "order_line",
&order_line_time_start->time_start);
                }

        }

        // rcint = bcp_batch(o_hdbc3);
        // if (rcint < 0)
        //      HandleErrorDBC(o_hdbc3);

        if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
        {
                rcint = bcp_done(o_hdbc3);
                if (rcint < 0)
                        HandleErrorDBC(o_hdbc3);

                SQLFreeStmt(o_hstmt3, SQL_DROP);
                SQLDisconnect(o_hdbc3);
                SQLFreeConnect(o_hdbc3);

                // if build index after load...
                if ((aptr->build_index == 1) && (aptr->index_order == 0))
                        BuildIndex("idxodlcl");

        }

}

//============================================================================
//
// Function   : GetPermutation
//
//============================================================================

void GetPermutation(int perm[], int n)
{
        int i, r, t;

        for (i=1;i<=n;i++)
                perm[i] = i;

        for (i=1;i<=n;i++)
        {
                r = RandomNumber(i,n);
                t = perm[i];
                perm[i] = perm[r];
                perm[r] = t;
        }

}


//============================================================================
//
// Function   : CheckForCommit
//
//============================================================================

void CheckForCommit(HDBC hdbc,
                                                HSTMT hstmt,
                                        int rows_loaded,
                                                char *table_name,
                                        long *time_start)
{

   long         time_end, time_diff;
        // DBINT       rcint;

   if ( !(rows_loaded % aptr->batch) )
   {

                // rcint = bcp_batch(hdbc);
                // if (rcint < 0)
                //      HandleErrorDBC(hdbc);

                time_end = (TimeNow() / MILLI);
                time_diff = time_end - *time_start;

                printf("-> Loaded %ld rows into %s in %ld sec - Total = %d (%.2f rps)\n",
                        aptr->batch,
                        table_name,
                                time_diff,
                        rows_loaded,
                                (float) aptr->batch / (time_diff ? time_diff : 1L));
```

```
                        *time_start = time_end;
    }

    return;
}


//============================================================================
//
// Function    : OpenConnections
//
//============================================================================

void OpenConnections()
{
        RETCODE     rc;

        char                        szDriverString[300];
        char                        szDriverStringOut[1024];
        SQLSMALLINT                 cbDriverStringOut;


        SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv );

        SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0 );

        SQLAllocHandle(SQL_HANDLE_DBC, henv , &i_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &w_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc2);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc2);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc3);

        SQLSetConnectAttr(i_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(w_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(c_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(c_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );

        // Open connections to SQL Server

        // Connection 1

        sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                        aptr->server,
                                                        aptr->user,
                                                        aptr->password,
                                                        aptr->database );

        rc = SQLSetConnectOption (i_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);

if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

rc = SQLDriverConnect ( i_hdbc1,
                                                NULL,
                                                (SQLCHAR*)&szDriverString[0] ,
                                                SQL_NTS,
                                                (SQLCHAR*)&szDriverStringOut[0],
                                                sizeof(szDriverStringOut),
                                                &cbDriverStringOut,
                                                SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

// Connection 2

sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                        aptr->server,
                                                        aptr->user,
                                                        aptr->password,
                                                        aptr->database );

rc = SQLSetConnectOption (w_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = SQLDriverConnect ( w_hdbc1,
                                                        NULL,
                                                        (SQLCHAR*)&szDriverString[0] ,
                                                        SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],
                                                        sizeof(szDriverStringOut),
                                                        &cbDriverStringOut,
                                                        SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

// Connection 3

sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                        aptr->server,
                                                        aptr->user,
                                                        aptr->password,
                                                        aptr->database );

rc = SQLSetConnectOption (c_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc1);

rc = SQLDriverConnect ( c_hdbc1,
                                                        NULL,
                                                        (SQLCHAR*)&szDriverString[0] ,
                                                        SQL_NTS,
```

```
(SQLCHAR*)&szDriverStringOut[0],
                                                        sizeof(szDriverStringOut),
                                                        &cbDriverStringOut,
                                                        SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc1);

// Connection 4

sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                        aptr->server,
                                                        aptr->user,
                                                        aptr->password,
                                                        aptr->database );

rc = SQLSetConnectOption (c_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

rc = SQLDriverConnect ( c_hdbc2,
                                                        NULL,
                                                        (SQLCHAR*)&szDriverString[0] ,
                                                        SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],
                                                        sizeof(szDriverStringOut),
                                                        &cbDriverStringOut,
                                                        SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

// Connection 5

sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                        aptr->server,
                                                        aptr->user,
                                                        aptr->password,
                                                        aptr->database );

rc = SQLSetConnectOption (o_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

rc = SQLDriverConnect ( o_hdbc1,
                                                        NULL,
                                                        (SQLCHAR*)&szDriverString[0] ,
                                                        SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],
                                                        sizeof(szDriverStringOut),
                                                        &cbDriverStringOut,
                                                        SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

// Connection 6

sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                        aptr->server,
                                                        aptr->user,
                                                        aptr->password,
                                                        aptr->database );

rc = SQLSetConnectOption (o_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

rc = SQLDriverConnect ( o_hdbc2,
                                                        NULL,
                                                        (SQLCHAR*)&szDriverString[0] ,
                                                        SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],
                                                        sizeof(szDriverStringOut),
                                                        &cbDriverStringOut,
                                                        SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

// Connection 7

sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                        aptr->server,
                                                        aptr->user,
                                                        aptr->password,
                                                        aptr->database );

rc = SQLSetConnectOption (o_hdbc3, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

rc = SQLDriverConnect ( o_hdbc3,
                                                        NULL,
                                                        (SQLCHAR*)&szDriverString[0] ,
                                                        SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],
                                                        sizeof(szDriverStringOut),
                                                        &cbDriverStringOut,
                                                        SQL_DRIVER_NOPROMPT );

if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

}

//==========================================================================
//
// Function name: BuildIndex
```

```
//
//============================================================================

void BuildIndex(char            *index_script)
{
        char            cmd[256];

        printf("Starting index creation:  %s\n",index_script);

        sprintf(cmd, "isql -S%s -U%s -P%s -e -i%s\\%s.sql > logs\\%s.log",
                                                aptr->server,
                                                aptr->user,
                                                aptr->password,
                                                aptr->index_script_path,
                                                index_script,
                                                index_script);

        system(cmd);

        printf("Finished index creation:  %s\n",index_script);
}


void HandleErrorDBC (SQLHDBC  hdbc1)
{

        SQLCHAR                 SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
        SQLINTEGER  NativeError;
        SQLSMALLINT i, MsgLen;
        SQLRETURN   rc2;
        char                    timebuf[128];
        char                    datebuf[128];
        FILE                    *fp1;

        i = 1;
        while (( rc2 = SQLGetDiagRec(SQL_HANDLE_DBC , hdbc1, i, SqlState , &NativeError,
                                                Msg, sizeof(Msg) , &MsgLen )) != SQL_NO_DATA )
        {

                sprintf( szLastError , "%s" ,  Msg );

                _strtime(timebuf);
                _strdate(datebuf);

                printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

                fp1 = fopen("logs\\tpccldr.err","w");
                if (fp1 == NULL)
                        printf("ERROR:  Unable to open errorlog file.\n");
                else
                {
                        fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf, szLastError);
                        fclose(fp1);
                }

                i++;

        }
}

void HandleErrorSTMT (HSTMT  hstmt1)
{

        SQLCHAR                 SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
        SQLINTEGER  NativeError;
        SQLSMALLINT i, MsgLen;
        SQLRETURN   rc2;
        char                    timebuf[128];
        char                    datebuf[128];
        FILE                    *fp1;

        i = 1;
        while (( rc2 = SQLGetDiagRec(SQL_HANDLE_STMT , hstmt1, i, SqlState , &NativeError,
                                                Msg, sizeof(Msg) , &MsgLen )) != SQL_NO_DATA )
        {

                sprintf( szLastError , "%s" ,  Msg );

                _strtime(timebuf);
                _strdate(datebuf);

                printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

                fp1 = fopen("logs\\tpccldr.err","w");
                if (fp1 == NULL)
                        printf("ERROR:  Unable to open errorlog file.\n");
                else
                {
                        fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf, szLastError);
                        fclose(fp1);
                }

                i++;

        }
}


void FormatDate ( char* szTimeCOutput )
{

        struct tm when;
        time_t now;

        time( &now );
        when  = *localtime( &now );

        mktime( &when );

        // odbc datetime format
```

```
		strftime( szTimeCOutput , 30 , "%Y-%m-%d %H:%M:%S.000", &when );

		return;

}

//============================================================================
//
// Function   : CheckDataBase
//
//============================================================================

void CheckDataBase()
{

		RETCODE		rc;

		char				szDriverString[300];
		char				szDriverStringOut[1024];
		char				TablesBitMap[9] = {"000000000"};
		int					i, ExitFlag;

		SQLSMALLINT			cbDriverStringOut;
		SQLCHAR				TabName[10];
		SQLINTEGER			TabNameInd, TabCount, TabCountInd;


		ExitFlag = 0;

		SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv );

		SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0 );

		SQLAllocHandle(SQL_HANDLE_DBC, henv , &v_hdbc);

		SQLSetConnectAttr(v_hdbc, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );

		// Open connection to SQL Server

		sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
								aptr->server,
								aptr->user,
								aptr->password,
								aptr->database );

		rc = SQLSetConnectAttr( v_hdbc, SQL_ATTR_PACKET_SIZE, (SQLPOINTER)aptr->pack_size,
SQL_IS_UINTEGER );
		if (rc != SQL_SUCCESS)
			HandleErrorDBC(v_hdbc);

		rc = SQLDriverConnect ( v_hdbc,
					NULL,
					(SQLCHAR*)&szDriverString[0] ,
					SQL_NTS,
					(SQLCHAR*)&szDriverStringOut[0],
					sizeof(szDriverStringOut),
					&cbDriverStringOut,
					SQL_DRIVER_NOPROMPT );

		// if the rc is SQL_ERROR, the the TPCC database probably does not exist
		if (rc == SQL_ERROR)
		{
			printf("The database TPCC does not appear to exist!\n");
			printf("\nCheck LOGS\\ directory for database creation errors.\n");

			// cleanup database connections and handles
			SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
			SQLDisconnect(v_hdbc);
			SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

			// since there is not a database, exit back to SETUP.CMD
			exit(1);
		}

		if ( SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt) != SQL_SUCCESS )
			HandleErrorDBC(v_hdbc);

		if ( SQLBindCol(v_hstmt, 1, SQL_C_ULONG, &TabCount, 0, &TabCountInd) != SQL_SUCCESS )
			HandleErrorSTMT(v_hstmt);

		// count the number of user tables from sysobjects
		rc = SQLExecDirect(v_hstmt, "select count(*) from sysobjects where xtype =\'U\'", SQL_NTS);
		if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
			HandleErrorSTMT(v_hstmt);

		if ( SQLFetch(v_hstmt) != SQL_SUCCESS )
			HandleErrorSTMT(v_hstmt);

		// if the number of tables is less than 9, select all the user tables in TPCC
		if (TabCount != 9)
		{
			SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);

			SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt);

			if ( SQLBindCol(v_hstmt, 1, SQL_C_CHAR, &TabName, sizeof(TabName), &TabNameInd) !=
SQL_SUCCESS )
				HandleErrorSTMT(v_hstmt);

			// select the list of user tables into a result set
			rc = SQLExecDirect(v_hstmt, "select * from sysobjects where xtype =\'U\'", SQL_NTS);
			if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
				HandleErrorSTMT(v_hstmt);

			// go through the result set and set the bitmap for each found table
			// set the bitmap to '1' if the table name is found

			while ((rc = SQLFetch(v_hstmt)) != SQL_NO_DATA)
			{
				switch( TabName[0] )
```

```
		{
		case 'w':
			TablesBitMap[0] = '1';
			break;
		case 'd':
			TablesBitMap[1] = '1';
			break;
		case 'c':
			TablesBitMap[2] = '1';
			break;
		case 'h':
			TablesBitMap[3] = '1';
			break;
		case 'n':
			TablesBitMap[4] = '1';
			break;
		case 'o':
			if (TabName[5] = 's')
				TablesBitMap[5] = '1';
			if (TabName[5] = '_')
				TablesBitMap[6] = '1';
			break;
		case 'i':
			TablesBitMap[7] = '1';
			break;
		case 's':
			TablesBitMap[8] = '1';
			break;
		}
}

// a '0' ExitFlag means do NOT exit the loader early, a '1' means exit the loader early
ExitFlag = 0;

// interate through the bitmap to display which table(s) is actually missing
for (i = 0; i <= 8; i++)
{
		switch(i)
		{
		case 0:
			if (TablesBitMap[i] == '0')
			{
				printf("The Warehouse table is missing or damaged.\n");
				ExitFlag = 1;
			}
			break;
		case 1:
			if (TablesBitMap[i] == '0')
			{
				printf("The District table is missing or damaged.\n");
				ExitFlag = 1;
			}
			break;
		case 2:
			if (TablesBitMap[i] == '0')
			{
				printf("The Customer table is missing or damaged.\n");
				ExitFlag = 1;
			}
			break;
		case 3:
			if (TablesBitMap[i] == '0')
			{
				printf("The History table is missing or damaged.\n");
				ExitFlag = 1;
			}
			break;
		case 4:
			if (TablesBitMap[i] == '0')
			{
				printf("The New_Order table is missing or damaged.\n");
				ExitFlag = 1;
			}
			break;
		case 5:
			if (TablesBitMap[i] == '0')
			{
				printf("The Orders table is missing or damaged.\n");
				ExitFlag = 1;
			}
			break;
		case 6:
			if (TablesBitMap[i] == '0')
			{
				printf("The Order_Line table is missing or damaged.\n");
				ExitFlag = 1;
			}
			break;
		case 7:
			if (TablesBitMap[i] == '0')
			{
				printf("The Item table is missing or damaged.\n");
				ExitFlag = 1;
			}
			break;
		case 8:
			if (TablesBitMap[i] == '0')
			{
				printf("The Stock table is missing or damaged.\n");
				ExitFlag = 1;
			}
			break;
		}
}

// if one or more tables are missing, display message and exit the loader
if (ExitFlag = 1)
{
		printf("\nExiting TPC-C Loader!\n");
		printf("\nCheck LOGS\\ directory for database\n");
```

```
                printf("or table creation errors.\n");

                // cleanup database connections and handles
                SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
                SQLDisconnect(v_hdbc);
                SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

                exit(1);
            }
        }

        // cleanup database connections and handles
        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
        SQLDisconnect(v_hdbc);
        SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

        return;
}


GETARGS.C
//      File:           GETARGS.C
//                              Microsoft TPC-C Kit Ver. 4.21
//                              Copyright Microsoft, 1996, 1997, 1998, 1999, 2000
//      Purpose:        Source file for command line processing


// Includes
#include "tpcc.h"

//===============================================================================
//
// Function name: GetArgsLoader
//
//===============================================================================

void GetArgsLoader(int argc, char **argv, TPCCLDR_ARGS *pargs)
{
   int          i;
   char         *ptr;

#ifdef DEBUG
   printf("[%ld]DBG: Entering GetArgsLoader()\n", (int) GetCurrentThreadId());
#endif

   /* init args struct with some useful values */
   pargs->server                = SERVER;
   pargs->user                  = USER;
   pargs->password              = PASSWORD;
   pargs->database              = DATABASE;
   pargs->batch                 = BATCH;
   pargs->num_warehouses                = UNDEF;
        pargs->tables_all                       = TRUE;
        pargs->table_item                       = FALSE;
        pargs->table_warehouse          = FALSE;
        pargs->table_customer           = FALSE;
        pargs->table_orders                     = FALSE;
        pargs->loader_res_file          = LOADER_RES_FILE;
        pargs->pack_size                = DEFLDPACKSIZE;
        pargs->starting_warehouse       = DEF_STARTING_WAREHOUSE;
        pargs->build_index                      = BUILD_INDEX;
        pargs->index_order                      = INDEX_ORDER;
        pargs->index_script_path    = INDEX_SCRIPT_PATH;
        pargs->scale_down                       = SCALE_DOWN;

/* check for zero command line args */
if ( argc == 1 )
                GetArgsLoaderUsage();

for (i = 1; i < argc; ++i)
{
                if (argv[i][0] != '-' && argv[i][0] != '/')
                {
                printf("\nUnrecognized command");
                GetArgsLoaderUsage();
                exit(1);
                }

                ptr = argv[i];

                switch (ptr[1])
                {
                case 'h':    /* Fall throught */
                case 'H':
                                GetArgsLoaderUsage();
                                break;

                case 'D':
                                pargs->database = ptr+2;
                                break;

                case 'P':
                                pargs->password = ptr+2;
                                break;

                case 'S':
                                pargs->server = ptr+2;
                                break;

                case 'U':
                                pargs->user = ptr+2;
                                break;

                case 'b':
                                pargs->batch = atol(ptr+2);
                                break;

                case 'W':
                                pargs->num_warehouses = atol(ptr+2);
                                break;
```

```
case 's':
                pargs->starting_warehouse = atol(ptr+2);
                break;

case 't':
                {
                        pargs->tables_all = FALSE;
                        if (strcmp(ptr+2,"item") == 0)
                                pargs->table_item = TRUE;
                        else if (strcmp(ptr+2,"warehouse") == 0)
                                pargs->table_warehouse = TRUE;
                        else if (strcmp(ptr+2,"customer") == 0)
                                pargs->table_customer = TRUE;
                        else if (strcmp(ptr+2,"orders") == 0)
                                pargs->table_orders = TRUE;
                        else
                        {
                        printf("\nUnrecognized command");
                        GetArgsLoaderUsage();
                        exit(1);
                        }

                        break;
                }

        case 'f':
                pargs->loader_res_file = ptr+2;
                break;

        case 'p':
                pargs->pack_size = atol(ptr+2);
                break;

        case 'i':
                pargs->build_index = atol(ptr+2);
                break;

        case 'o':
                pargs->index_order = atol(ptr+2);
                break;

        case 'c':
                pargs->scale_down = atol(ptr+2);
                break;

case 'd':
                pargs->index_script_path = ptr+2;
                break;

default:
        GetArgsLoaderUsage();
        exit(-1);
        break;
}
```

```
    }

    /* check for required args */
    if (pargs->num_warehouses == UNDEF )
    {
                printf("Number of Warehouses is required\n");
                exit(-2);
    }

    return;
}


//==============================================================================
//
// Function name: GetArgsLoaderUsage
//
//==============================================================================

void GetArgsLoaderUsage()
{

#ifdef DEBUG
    printf("[%ld]DBG: Entering GetArgsLoaderUsage()\n", (int) GetCurrentThreadId());
#endif


    printf("TPCCLDR:\n\n");
        printf("Parameter                                Default\n");
        printf("-------------------------------------------------------------\n");
    printf("-W Number of Warehouses to Load          Required \n");
    printf("-S Server                                %s\n", SERVER);
    printf("-U Username                              %s\n", USER);
    printf("-P Password                              %s\n", PASSWORD);
    printf("-D Database                              %s\n", DATABASE);
        printf("-b Batch Size                            %ld\n", (long) BATCH);
        printf("-p TDS packet size                       %ld\n", (long) DEFLDPACKSIZE);
        printf("-f Loader Results Output Filename        %s\n", LOADER_RES_FILE);
        printf("-s Starting Warehouse                    %ld\n", (long) DEF_STARTING_WAREHOUSE);
        printf("-i Build Option (data = 0, data and index = 1)   %ld\n", (long) BUILD_INDEX);
        printf("-o Cluster Index Build Order (before = 1, after = 0)  %ld\n", (long) INDEX_ORDER);
        printf("-c Build Scaled Database (normal = 0, tiny = 1)   %ld\n", (long) SCALE_DOWN);
        printf("-d Index Script Path                     %s\n", INDEX_SCRIPT_PATH);
        printf("-t Table to Load                         all tables \n");
    printf("   [item|warehouse|customer|orders]\n");
    printf("   Notes: \n");
    printf("     - the '-t' parameter may be included multiple times to \n");
    printf("      specify multiple tables to be loaded \n");
    printf("     - 'item' loads ITEM table \n");
    printf("     - 'warehouse' loads WAREHOUSE, DISTRICT, and STOCK tables \n");
    printf("     - 'customer' loads CUSTOMER and HISTORY tables \n");
    printf("     - 'orders' load NEW-ORDER, ORDERS, ORDER-LINE tables \n");

        printf("\nNote:  Command line switches are case sensitive.\n");
```

```
    exit(0);
}
```

RANDOM.C

```
//              File:                   RANDOM.C
//                                      Microsoft TPC-C Kit Ver. 4.21
//                                      Copyright Microsoft, 1996, 1997, 1998, 1999, 2000
//              Purpose:        Random number generation routines for database loader


// Includes
#include "tpcc.h"
#include "math.h"

// Defines
#define A           16807
#define M       2147483647
#define Q           127773      /* M div A */
#define R            2836       /* M mod A */
#define Thread                  __declspec(thread)

// Globals
long            Thread Seed = 0;        /* thread local seed */

/*****************************************************************************
*                                                                            *
* random -                                                                   *
*       Implements a GOOD pseudo random number generator.  This generator    *
*       will/should? run the complete period before repeating.               *
*                                                                            *
* Copied from:                                                               *
*       Random Numbers Generators: Good Ones Are Hard to Find.               *
*       Communications of the ACM - October 1988 Volume 31 Number 10         *
*                                                                            *
* Machine Dependencies:                                                      *
*       long must be 2 ^ 31 - 1 or greater.                                  *
*                                                                            *
*****************************************************************************/

/*****************************************************************************
* seed - load the Seed value used in irand and drand.  Should be used before *
*       first call to irand or drand.                                        *
*****************************************************************************/

void seed(long val)
{

#ifdef DEBUG
    printf("[%ld]DBG: Entering seed()...\n", (int) GetCurrentThreadId());
                printf("Old Seed %ld New Seed %ld\n",Seed, val);
#endif

    if ( val < 0 )
                val = abs(val);

    Seed = val;

}


/*****************************************************************************
*                                                                            *
* irand - returns a 32 bit integer pseudo random number with a period of     *
*       1 to 2 ^ 32 - 1.                                                     *
*                                                                            *
* parameters:                                                                *
*       none.                                                                *
*                                                                            *
* returns:                                                                   *
*       32 bit integer - defined as long ( see above ).                      *
*                                                                            *
* side effects:                                                              *
*       seed get recomputed.                                                 *
*****************************************************************************/

long irand()
{
    register long   s;      /* copy of seed */
    register long   test;   /* test flag */
    register long   hi;     /* tmp value for speed */
    register long   lo;     /* tmp value for speed */

#ifdef DEBUG
    printf("[%ld]DBG: Entering irand()...\n", (int) GetCurrentThreadId());
#endif

    s = Seed;
    hi = s / Q;
    lo = s % Q;

    test = A * lo - R * hi;
    if ( test > 0 )
                                Seed = test;
    else
                                Seed = test + M;

    return( Seed );
}


/*****************************************************************************
*                                                                            *
* drand - returns a double pseudo random number between 0.0 and 1.0.         *
*       See irand.                                                           *
*****************************************************************************/
```

```
double drand()
{

#ifdef DEBUG
   printf("[%ld]DBG: Entering drand()...\n", (int) GetCurrentThreadId());
#endif

   return( (double)irand() / 2147483647.0);
}


//====================================================================
// Function   : RandomNumber
//
// Description:
//====================================================================
long RandomNumber(long lower, long upper)
{
   long rand_num;

#ifdef DEBUG
   printf("[%ld]DBG: Entering RandomNumber()...\n", (int) GetCurrentThreadId());
#endif

                  if ( upper == lower )            /* pgd 08-13-96 perf enhancement */
                                 return lower;

                  upper++;

   if ( upper <= lower )
                                 rand_num = upper;
   else
                                 rand_num = lower + irand() % (upper - lower); /* pgd 08-13-96 perf enhancement */


#ifdef DEBUG
   printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                                                (int) GetCurrentThreadId(), lower, upper, rand_num);
#endif


   return rand_num;
}


#if 0

//Orginal code pgd 08/13/96

long RandomNumber(long lower,
                                                long upper)
{
   long rand_num;

#ifdef DEBUG
   printf("[%ld]DBG: Entering RandomNumber()...\n", (int) GetCurrentThreadId());
#endif

                  upper++;

   if ((upper <= lower))
                                 rand_num = upper;
   else
                                 rand_num = lower + irand() % ((upper > lower) ? upper - lower : upper);


#ifdef DEBUG
   printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                                                (int) GetCurrentThreadId(), lower, upper, rand_num);
#endif


   return rand_num;
}
#endif

//====================================================================
// Function   : NURand
//
// Description:
//====================================================================
long NURand(int iConst,
                  long x,
                  long y,
                  long C)
{
   long rand_num;

#ifdef DEBUG
   printf("[%ld]DBG: Entering NURand()...\n", (int) GetCurrentThreadId());
#endif

   rand_num = (((RandomNumber(0,iConst) | RandomNumber(x,y)) + C) % (y-x+1))+x;

#ifdef DEBUG
   printf("[%ld]DBG: NURand: num = %d\n", (int) GetCurrentThreadId(), rand_num);
#endif

   return rand_num;
}
```

STRINGS.C

```
//          File:          STRINGS.C
//                         Microsoft TPC-C Kit Ver. 4.21
//                         Copyright Microsoft, 1996, 1997, 1998, 1999, 2000
```

APPENDIX B – DATABASE DESIGN

```
//              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

C.1 Microsoft Windowns 2000 Datacenter Server Configuration

The following services were disabled in the Windows 2000 Control Panel/Services:

?? Alerter
?? Computer Browser
?? Distributed File System
?? Distributed Link Tracking Client
?? DNS Client
?? Global Array Manager Server
?? IPSEC Policy Agent
?? License Logging Service
?? Messenger
?? Microsoft Search
?? Print Spooler
?? Process Control Service
?? Remote Registry Service
?? Removable Storage
?? Run as Service
?? Task scheduler
?? World Wide Web Publishing Service

C.2 Server System Configuration Parameters

The following registry key was added to disable the kernel counters for Global and Per-Process I/Os:
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\I/O System] "CountOperations"=dword:00000000

Microsoft Windows 2000 Datacenter Server System Information Report For LXr8500DC

System Information report written at: 12/19/2000 06:20:51 PM
[System Information]

[Following are sub-categories of this main category]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Advanced Server
Version	5.0.2195 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	PRF_SUT6
System Manufacturer	HP
System Model	HP NetServer LXr 8500
System Type	X86-based PC
Processor	x86 Family 6 Model 10 Stepping 0 GenuineIntel ~700 Mhz
Processor	x86 Family 6 Model 10 Stepping 0 GenuineIntel ~700 Mhz
Processor	x86 Family 6 Model 10 Stepping 0 GenuineIntel ~700 Mhz
Processor	x86 Family 6 Model 10 Stepping 0 GenuineIntel ~700 Mhz
Processor	x86 Family 6 Model 10 Stepping 0 GenuineIntel ~700 Mhz
Processor	x86 Family 6 Model 10 Stepping 0 GenuineIntel ~700 Mhz
Processor	x86 Family 6 Model 10 Stepping 0 GenuineIntel ~700 Mhz
Processor	x86 Family 6 Model 10 Stepping 0 GenuineIntel ~700 Mhz
BIOS Version	OCPRF100- PhoenixBIOS 4.0 Release 6.0
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition1
Locale	United States
User Name	PRF_SUT6\Administrator
Time Zone	Pacific Standard Time
Total Physical Memory	7,863,616 KB
Available Physical Memory	7,681,088 KB
Total Virtual Memory	16,301,960 KB
Available Virtual Memory	16,020,620 KB
Page File Space	8,438,344 KB
Page File	C:\pagefile.sys

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

Resource	Device
IRQ 10	Compaq PCI Hotplug Controller
IRQ 10	Compaq PCI Hotplug Controller
IRQ 10	Compaq PCI Hotplug Controller
IRQ 10	Compaq PCI Hotplug Controller

[DMA]

Channel	Device	Status
4	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
0x0000-0x0CF7	PCI bus	OK
0x0000-0x0CF7	Direct memory access controller	OK
0x0000-0x0CF7	PCI bus	OK
0x0D00-0x2000	PCI bus	OK
0x4500-0xFFFF	PCI bus	OK
0x1000-0x10FF	Hewlett Packard NetRAID-4M RAID Controller	OK
0x1C00-0x1C1F	HP NetServer 10/100TX PCI LAN Adapter	OK
0x1400-0x14FF	Symbios Logic 896, 22910 PCI SCSI Adapter	OK
0x1800-0x18FF	Symbios Logic 896, 22910 PCI SCSI Adapter	OK
0x03B0-0x03BB	Cirrus Logic 5446 Compatible Graphics Adapter	OK
0x03C0-0x03DF	Cirrus Logic 5446 Compatible Graphics Adapter	OK
0x0A79-0x0A79	ISAPNP Read Data Port	OK
0x0279-0x0279	ISAPNP Read Data Port	OK
0x0274-0x0277	ISAPNP Read Data Port	OK
0x00B3-0x00B3	Motherboard resources	OK
0x0C10-0x0C3F	Motherboard resources	OK
0x0CA8-0x0CAF	Motherboard resources	OK
0x0CC0-0x0CCF	Motherboard resources	OK
0x0010-0x001F	Direct memory access controller	OK
0x0080-0x009F	Direct memory access controller	OK
0x00C0-0x00DF	Direct memory access controller	OK
0x0070-0x0077	System CMOS/real time clock	OK
0x0020-0x0021	Programmable interrupt controller	OK
0x0024-0x0025	Programmable interrupt controller	OK
0x0028-0x0029	Programmable interrupt controller	OK
0x002C-0x002D	Programmable interrupt controller	OK
0x0030-0x0031	Programmable interrupt controller	OK
0x0034-0x0035	Programmable interrupt controller	OK
0x0038-0x0039	Programmable interrupt controller	OK
0x003C-0x003D	Programmable interrupt controller	OK
0x00A0-0x00A1	Programmable interrupt controller	OK
0x00A4-0x00A5	Programmable interrupt controller	OK
0x00A8-0x00A9	Programmable interrupt controller	OK
0x00AC-0x00AD	Programmable interrupt controller	OK
0x00B0-0x00B1	Programmable interrupt controller	OK
0x00B4-0x00B5	Programmable interrupt controller	OK
0x00B8-0x00B9	Programmable interrupt controller	OK

APPENDIX C – TUNABLE PARAMETERS

0x00BC-0x00BD	Programmable interrupt controller	OK
0x04D0-0x04D1	Programmable interrupt controller	OK
0x00F0-0x00FF	Numeric data processor	OK
0x0040-0x0043	System timer	OK
0x0050-0x0053	System timer	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
0x03F2-0x03F5	Standard floppy disk controller	OK
0x03F7-0x03F7	Standard floppy disk controller	OK
0x03F8-0x03FF	Communications Port (COM1)	OK
0x02F8-0x02FF	Communications Port (COM2)	OK
0x1C40-0x1C4F	Intel(r) 82371AB/EB PCI Bus Master IDE Controller	OK
0x01F0-0x01F7	Primary IDE Channel	OK
0x03F6-0x03F6	Primary IDE Channel	OK
0x1C20-0x1C3F	Intel 82371AB/EB PCI to USB Universal Host Controller	OK
0x2000-0x24FF	PCI bus	OK
0x2000-0x24FF	Hewlett Packard NetRAID-4M RAID Controller	OK
0x2400-0x24FF	Hewlett Packard NetRAID-4M RAID Controller	OK
0x3000-0x34FF	PCI bus	OK
0x3000-0x34FF	Hewlett Packard NetRAID-4M RAID Controller	OK
0x3400-0x34FF	Hewlett Packard NetRAID-4M RAID Controller	OK
0x4000-0x44FF	PCI bus	OK
0x4000-0x44FF	Hewlett Packard NetRAID-4M RAID Controller	OK
0x4400-0x44FF	Hewlett Packard NetRAID-4M RAID Controller	OK

[IRQs]

IRQ Number	Device
9	Microsoft ACPI-Compliant System
10	Compaq PCI Hotplug Controller
10	Compaq PCI Hotplug Controller
10	Compaq PCI Hotplug Controller
10	Compaq PCI Hotplug Controller
61	Hewlett Packard NetRAID-4M RAID Controller
54	HP NetServer 10/100TX PCI LAN Adapter
58	Symbios Logic 896, 22910 PCI SCSI Adapter
18	Symbios Logic 896, 22910 PCI SCSI Adapter
8	System CMOS/real time clock
13	Numeric data processor
1	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
12	Microsoft PS/2 Mouse
6	Standard floppy disk controller
4	Communications Port (COM1)
3	Communications Port (COM2)
14	Primary IDE Channel
49	Intel 82371AB/EB PCI to USB Universal Host Controller
44	Hewlett Packard NetRAID-4M RAID Controller
40	Hewlett Packard NetRAID-4M RAID Controller
32	Hewlett Packard NetRAID-4M RAID Controller
28	Hewlett Packard NetRAID-4M RAID Controller
24	Hewlett Packard NetRAID-4M RAID Controller
20	Hewlett Packard NetRAID-4M RAID Controller

[Memory]

Range	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	Cirrus Logic 5446 Compatible Graphics Adapter	OK
0xC8000-0xDFFFF	PCI bus	OK
0xE0000-0xFFFFF	PCI bus	OK
0xF0000000-0xFDFFFFFF	PCI bus	OK
0xFFF00000-0xFFFFFFFF	PCI bus	OK
0xFA007000-0xFA0070FF	Compaq PCI Hotplug Controller	OK
0xFA000000-0xFA001FFF	Hewlett Packard NetRAID-4M RAID Controller	OK
0xFA400000-0xFA400FFF	HP NetServer 10/100TX PCI LAN Adapter	OK
0xFA100000-0xFA1FFFFF	HP NetServer 10/100TX PCI LAN Adapter	OK
0xFA007400-0xFA0077FF	Symbios Logic 896, 22910 PCI SCSI Adapter	OK
0xFA002000-0xFA003FFF	Symbios Logic 896, 22910 PCI SCSI Adapter	OK
0xFA007800-0xFA007BFF	Symbios Logic 896, 22910 PCI SCSI Adapter	OK
0xFA004000-0xFA005FFF	Symbios Logic 896, 22910 PCI SCSI Adapter	OK
0xFC000000-0xFDFFFFFF	Cirrus Logic 5446 Compatible Graphics Adapter	OK
0xFA006000-0xFA006FFF	Cirrus Logic 5446 Compatible Graphics Adapter	OK
0xFE000000-0xFE3FFFFF	PCI bus	OK
0xFE000000-0xFE3FFFFF	Hewlett Packard NetRAID-4M RAID Controller	OK
0xFE004000-0xFE0040FF	Compaq PCI Hotplug Controller	OK
0xFE002000-0xFE003FFF	Hewlett Packard NetRAID-4M RAID Controller	OK
0xFE400000-0xFE7FFFFF	PCI bus	OK
0xFE400000-0xFE7FFFFF	Hewlett Packard NetRAID-4M RAID Controller	OK
0xFE404000-0xFE4040FF	Compaq PCI Hotplug Controller	OK
0xFE402000-0xFE403FFF	Hewlett Packard NetRAID-4M RAID Controller	OK
0xFE800000-0xFEBFFFFF	PCI bus	OK
0xFE800000-0xFEBFFFFF	Hewlett Packard NetRAID-4M RAID Controller	OK
0xFE804000-0xFE8040FF	Compaq PCI Hotplug Controller	OK
0xFE802000-0xFE803FFF	Hewlett Packard NetRAID-4M RAID Controller	OK

[Components]

[Following are sub-categories of this main category]

[Multimedia]

[Following are sub-categories of this main category]

[Audio Codecs]

Codec	Manufacturer	Description	Status	File	Version	Size	Creation Date
c:\winnt\system32\iac25_32.ax	Intel Corporation	Indeo® audio software	OK				
				C:\WINNT\System32\IAC25_32.AX	2.05.53	195.00 KB (199,680 bytes)	12/7/1999 4:00:00 AM
c:\winnt\system32\lhacm.acm	Microsoft Corporation		OK				
				C:\WINNT\System32\LHACM.ACM	4.4.3385	33.27 KB (34,064 bytes)	10/27/2000 4:43:52 PM
c:\winnt\system32\msg723.acm	Microsoft Corporation		OK				
				C:\WINNT\System32\MSG723.ACM	4.4.3385	106.77 KB (109,328 bytes)	10/27/2000 4:43:51 PM
c:\winnt\system32\tssoft32.acm	DSP GROUP, INC.		OK				
				C:\WINNT\System32\TSSOFT32.ACM	1.01	9.27 KB (9,488 bytes)	12/7/1999 4:00:00 AM
c:\winnt\system32\imaadp32.acm	Microsoft Corporation		OK				
				C:\WINNT\System32\IMAADP32.ACM	5.00.2134.1	16.27 KB (16,656 bytes)	12/7/1999 4:00:00 AM
c:\winnt\system32\msgsm32.acm	Microsoft Corporation		OK				
				C:\WINNT\System32\MSGSM32.ACM	5.00.2134.1	22.27 KB (22,800 bytes)	12/7/1999 4:00:00 AM
c:\winnt\system32\msadp32.acm	Microsoft Corporation		OK				
				C:\WINNT\System32\MSADP32.ACM	5.00.2134.1	14.77 KB (15,120 bytes)	12/7/1999 4:00:00 AM

APPENDIX C – TUNABLE PARAMETERS

c:\winnt\system32\msg711.acm Microsoft Corporation OK
C:\WINNT\System32\MSG711.ACM 5.00.2134.1 10.27 KB (10,512 bytes) 12/7/1999 4:00:00 AM

[Video Codecs]

Codec Manufacturer Description Status File Version Size Creation Date
c:\winnt\system32\ir50_32.dll Intel Corporation Indeo® video 5.10 OK
C:\WINNT\System32\IR50_32.DLL R.5.10.15.2.55 737.50 KB (755,200 bytes) 12/7/1999 4:00:00 AM
c:\winnt\system32\msh261.drv Microsoft Corporation OK
C:\WINNT\System32\MSH261.DRV 4.4.3385 163.77 KB (167,696 bytes) 10/27/2000 4:43:51 PM
c:\winnt\system32\msh263.drv Microsoft Corporation OK
C:\WINNT\System32\MSH263.DRV 4.4.3385 252.27 KB (258,320 bytes) 10/27/2000 4:43:19 PM
c:\winnt\system32\msvidc32.dll Microsoft Corporation OK
C:\WINNT\System32\MSVIDC32.DLL 5.00.2134.1 27.27 KB (27,920 bytes) 12/7/1999 4:00:00 AM
c:\winnt\system32\msrle32.dll Microsoft Corporation OK
C:\WINNT\System32\MSRLE32.DLL 5.00.2134.1 10.77 KB (11,024 bytes) 12/7/1999 4:00:00 AM
c:\winnt\system32\ir32_32.dll Intel(R) Corporation OK
C:\WINNT\System32\IR32_32.DLL Not Available 194.50 KB (199,168 bytes) 12/7/1999 4:00:00 AM
c:\winnt\system32\iccvid.dll Radius Inc. OK C:\WINNT\System32\ICCVID.DLL
1.10.0.6 108.00 KB (110,592 bytes) 12/7/1999 4:00:00 AM

[CD-ROM]

Item Value
Drive D:
Description CD-ROM Drive
Media Loaded False
Media Type CD-ROM
Name TORiSAN CD-ROM CDR_U241
Manufacturer (Standard CD-ROM drives)
Status OK
Transfer Rate Not Available
SCSI Target ID 0
PNP Device ID IDE\CDROMTORISAN_CD-ROM_CDR_U241_____1.03____\5&201331&0&0.0.0

[Sound Device]

Item Value
No sound devices

[Display]

Item Value
Name Cirrus Logic 5446 Compatible Graphics Adapter
PNP Device ID PCI\VEN_1013&DEV_00B8&SUBSYS_00B81013&REV_45\3&267A616A&0&60
Adapter Type Cirrus Logic 5446BE, Cirrus Logic compatible
Adapter Description Cirrus Logic 5446 Compatible Graphics Adapter
Adapter RAM 2.00 MB (2,097,152 bytes)
Installed Drivers vga.sys,cirrus.sys,vga256.dll,vga64k.dll
Driver Version 5.00.2146.1
INF File display.inf (cirrus section)
Color Planes 1
Color Table Entries 65536
Resolution 1024 x 768 x 75 hertz
Bits/Pixel 16

[Infrared]

Item Value
No infrared devices

[Input]

[Following are sub-categories of this main category]

[Keyboard]

Item Value
Description Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
Name Enhanced (101- or 102-key)
Layout 00000409
PNP Device ID ACPI\PNP0303\4&43B47AD&0
NumberOfFunctionKeys 12

[Pointing Device]

Item Value
Hardware Type Microsoft PS/2 Mouse
Number of Buttons 2
Status OK
PNP Device ID ACPI\PNP0F03\4&43B47AD&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] HP NetServer 10/100TX PCI LAN Adapter
Adapter Type Ethernet 802.3
Product Name HP NetServer 10/100TX PCI LAN Adapter
Installed True
PNP Device ID PCI\VEN_8086&DEV_1229&SUBSYS_10C3103C&REV_05\3&267A616A&0&28
Last Reset 12/18/2000 8:29:15 AM
Index 0
Service Name HPTX
IP Address 15.75.206.170
IP Subnet 255.255.248.0

APPENDIX C – TUNABLE PARAMETERS

Default IP Gateway 15.75.200.1
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:90:27:E7:05:E2
Service Name HPTX
IRQ Number 54
I/O Port 0x1C00-0x1C1F
Driver c:\winnt\system32\drivers\hptxnt5.sys (80144, 4.01.67.0000)

Name [00000001] RAS Async Adapter
Adapter Type Not Available
Product Name RAS Async Adapter
Installed True
PNP Device ID Not Available
Last Reset 12/18/2000 8:29:15 AM
Index 1
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Not Available

Name [00000002] WAN Miniport (L2TP)
Adapter Type Not Available
Product Name WAN Miniport (L2TP)
Installed True
PNP Device ID ROOT\MS_L2TPMINIPORT\0000
Last Reset 12/18/2000 8:29:15 AM
Index 2
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Rasl2tp
Driver c:\winnt\system32\drivers\rasl2tp.sys (50800, 5.00.2179.1)

Name [00000003] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Name WAN Miniport (PPTP)
Installed True
PNP Device ID ROOT\MS_PPTPMINIPORT\0000
Last Reset 12/18/2000 8:29:15 AM
Index 3
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30
Service Name PptpMiniport
Driver c:\winnt\system32\drivers\raspptp.sys (47856, 5.00.2160.1)

Name [00000004] Direct Parallel
Adapter Type Not Available
Product Name Direct Parallel
Installed True
PNP Device ID ROOT\MS_PTIMINIPORT\0000
Last Reset 12/18/2000 8:29:15 AM
Index 4
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Raspti
Driver c:\winnt\system32\drivers\raspti.sys (16880, 5.00.2146.1)

Name [00000005] WAN Miniport (IP)
Adapter Type Not Available
Product Name WAN Miniport (IP)
Installed True
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 12/18/2000 8:29:15 AM
Index 5
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name NdisWan
Driver c:\winnt\system32\drivers\ndiswan.sys (90768, 5.00.2184.1)

Name [00000006] WAN Miniport (Network Monitor)
Adapter Type Not Available
Product Name WAN Miniport (Network Monitor)
Installed True
PNP Device ID ROOT\MS_NDISWANBH\0000
Last Reset 12/18/2000 8:29:15 AM
Index 6

APPENDIX C – TUNABLE PARAMETERS

Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name NdisWan
Driver c:\winnt\system32\drivers\ndiswan.sys (90768, 5.00.2184.1)

[Protocol]

Item Value
Name MSAFD Tcpip [TCP/IP]
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 16 bytes
MaximumMessageSize 0 bytes
MessageOriented False
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData True
SupportsGracefulClosing True
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD Tcpip [UDP/IP]
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 16 bytes
MaximumMessageSize 65467 bytes
MessageOriented True
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting True

Name RSVP UDP Service Provider
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 16 bytes
MaximumMessageSize 65467 bytes
MessageOriented True
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption True
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting True

Name RSVP TCP Service Provider
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 16 bytes
MaximumMessageSize 0 bytes
MessageOriented False
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption True
SupportsExpeditedData True
SupportsGracefulClosing True
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{102FF395-B92C-459A-8815-7CD8C0E95064}] 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_{102FF395-B92C-459A-8815-7CD8C0E95064}] 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_{A80BB721-6873-4855-88AE-6039C5501E9D}] 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_{A80BB721-6873-4855-88AE-6039C5501E9D}] 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_{D825D0FD-09A8-46CC-93F4-F40B9EC3A7F8}] 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_{D825D0FD-09A8-46CC-93F4-F40B9EC3A7F8}] DATAGRAM 2
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

[WinSock]

Item Value
File c:\winnt\system32\winsock.dll
Version 3.10
Size 2.80 KB (2,864 bytes)

File c:\winnt\system32\wsock32.dll
Version 5.00.2152.1
Size 21.27 KB (21,776 bytes)

[Ports]

[Following are sub-categories of this main category]

[Serial]

Item Value
Name COM1
Status OK
PNP Device ID ACPI\PNP0501\1
Maximum Input Buffer Size 0
Maximum Output Buffer Size False
Settable Baud Rate True
Settable Data Bits True
Settable Flow Control True

Settable Parity True
Settable Parity Check True
Settable Stop Bits True
Settable RLSD True
Supports RLSD True
Supports 16 Bit Mode False
Supports Special Characters False
Baud Rate 9600
Bits/Byte 8
Stop Bits 1
Parity None
Busy 0
Abort Read/Write on Error 0
Binary Mode Enabled -1
Continue XMit on XOff 0
CTS Outflow Control 0
Discard NULL Bytes 0
DSR Outflow Control 0
DSR Sensitivity 0
DTR Flow Control Type Enable
EOF Character 0
Error Replace Character 0
Error Replacement Enabled 0
Event Character 0
Parity Check Enabled 0
RTS Flow Control Type Enable
XOff Character 19
XOffXMit Threshold 512
XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control 0
IRQ Number 4
I/O Port 0x03F8-0x03FF
Driver c:\winnt\system32\drivers\serial.sys (62448, 5.00.2134.1)

Name COM2
Status OK
PNP Device ID ACPI\PNP0501\2
Maximum Input Buffer Size 0
Maximum Output Buffer Size False
Settable Baud Rate True
Settable Data Bits True
Settable Flow Control True
Settable Parity True
Settable Parity Check True
Settable Stop Bits True
Settable RLSD True
Supports RLSD True
Supports 16 Bit Mode False
Supports Special Characters False
Baud Rate 9600
Bits/Byte 8
Stop Bits 1
Parity None
Busy 0
Abort Read/Write on Error 0
Binary Mode Enabled -1
Continue XMit on XOff 0
CTS Outflow Control 0
Discard NULL Bytes 0
DSR Outflow Control 0
DSR Sensitivity 0
DTR Flow Control Type Enable
EOF Character 0
Error Replace Character 0
Error Replacement Enabled 0
Event Character 0
Parity Check Enabled 0
RTS Flow Control Type Enable
XOff Character 19
XOffXMit Threshold 512
XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control 0
IRQ Number 3
I/O Port 0x02F8-0x02FF
Driver c:\winnt\system32\drivers\serial.sys (62448, 5.00.2134.1)

[Parallel]

Item Value
No parallel port information

[Storage]

[Following are sub-categories of this main category]

[Drives]

Item Value
Drive A:
Description 3 1/2 Inch Floppy Drive

Drive C:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 8.46 GB (9,088,901,120 bytes)
Free Space 5.40 GB (5,802,348,544 bytes)
Volume Name
Volume Serial Number 00D4459F
Partition Disk #0, Partition #0
Partition Size 8.46 GB (9,088,902,144 bytes)
Starting Offset 32256 bytes
Drive Description Disk drive
Drive Manufacturer (Standard disk drives)
Drive Model HP 9.10GB B 80-8C02 SCSI Disk Device
Drive BytesPerSector 512
Drive MediaLoaded True

APPENDIX C – TUNABLE PARAMETERS

Drive MediaType Fixed hard disk media
Drive Partitions 1
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 1
Drive SCSITargetId 11
Drive SectorsPerTrack 63
Drive Size 9097159680 bytes
Drive TotalCylinders 1106
Drive TotalSectors 17767890
Drive TotalTracks 282030
Drive TracksPerCylinder 255

Drive E:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #5, Partition #0
Partition Size 135.59 GB (145,589,518,336 bytes)
Starting Offset 16384 bytes
Drive Description \\.\PHYSICALDRIVE5
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 1
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 5
Drive SCSITargetId 0
Drive SectorsPerTrack 32
Drive Size 145590583296 bytes
Drive TotalCylinders 138846
Drive TotalSectors 284356608
Drive TotalTracks 8886144
Drive TracksPerCylinder 64

Drive F:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #1, Partition #0
Partition Size 39.06 GB (41,943,023,616 bytes)
Starting Offset 16384 bytes
Drive Description \\.\PHYSICALDRIVE1
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 3
Drive SCSITargetId 0
Drive SectorsPerTrack 32
Drive Size 64424509440 bytes
Drive TotalCylinders 61440
Drive TotalSectors 125829120
Drive TotalTracks 3932160
Drive TracksPerCylinder 64

Drive G:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #3, Partition #0
Partition Size 39.06 GB (41,943,023,616 bytes)
Starting Offset 16384 bytes
Drive Description \\.\PHYSICALDRIVE3
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 4
Drive SCSITargetId 0
Drive SectorsPerTrack 32
Drive Size 64424509440 bytes
Drive TotalCylinders 61440
Drive TotalSectors 125829120
Drive TotalTracks 3932160
Drive TracksPerCylinder 64

Drive H:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #6, Partition #0
Partition Size 54.69 GB (58,720,239,616 bytes)
Starting Offset 16384 bytes
Drive Description \\.\PHYSICALDRIVE6

APPENDIX C – TUNABLE PARAMETERS

Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 6
Drive SCSITargetId 0
Drive SectorsPerTrack 32
Drive Size 655157624832 bytes
Drive TotalCylinders 624807
Drive TotalSectors 1279604736
Drive TotalTracks 39987648
Drive TracksPerCylinder 64

Drive I:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #7, Partition #0
Partition Size 39.06 GB (41,943,023,616 bytes)
Starting Offset 16384 bytes
Drive Description \\.\PHYSICALDRIVE7
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 7
Drive SCSITargetId 0
Drive SectorsPerTrack 32
Drive Size 64424509440 bytes
Drive TotalCylinders 61440
Drive TotalSectors 125829120
Drive TotalTracks 3932160
Drive TracksPerCylinder 64

Drive J:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #9, Partition #0
Partition Size 39.06 GB (41,943,023,616 bytes)
Starting Offset 16384 bytes
Drive Description \\.\PHYSICALDRIVE9
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 8
Drive SCSITargetId 0
Drive SectorsPerTrack 32
Drive Size 64424509440 bytes
Drive TotalCylinders 61440
Drive TotalSectors 125829120
Drive TotalTracks 3932160
Drive TracksPerCylinder 64

Drive K:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #11, Partition #0
Partition Size 54.69 GB (58,720,239,616 bytes)
Starting Offset 16384 bytes
Drive Description \\.\PHYSICALDRIVE11
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 9
Drive SCSITargetId 0
Drive SectorsPerTrack 32
Drive Size 436743438336 bytes
Drive TotalCylinders 416511
Drive TotalSectors 853014528
Drive TotalTracks 26656704
Drive TracksPerCylinder 64

Drive O:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available

APPENDIX C – TUNABLE PARAMETERS

Partition Disk #1, Partition #1
Partition Size 20.94 GB (22,480,420,864 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALDRIVE1
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 3
Drive SCSITargetId 0
Drive SectorsPerTrack 32
Drive Size 64424509440 bytes
Drive TotalCylinders 61440
Drive TotalSectors 125829120
Drive TotalTracks 3932160
Drive TracksPerCylinder 64

Drive P:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #3, Partition #1
Partition Size 20.94 GB (22,480,420,864 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALDRIVE3
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 4
Drive SCSITargetId 0
Drive SectorsPerTrack 32
Drive Size 64424509440 bytes
Drive TotalCylinders 61440
Drive TotalSectors 125829120
Drive TotalTracks 3932160
Drive TracksPerCylinder 64

Drive Q:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #6, Partition #1
Partition Size 26.37 GB (28,311,552,000 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALDRIVE6
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 6
Drive SCSITargetId 0
Drive SectorsPerTrack 32
Drive Size 655157624832 bytes
Drive TotalCylinders 624807
Drive TotalSectors 1279604736
Drive TotalTracks 39987648
Drive TracksPerCylinder 64

Drive R:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #7, Partition #1
Partition Size 20.94 GB (22,480,420,864 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALDRIVE7
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 7
Drive SCSITargetId 0
Drive SectorsPerTrack 32
Drive Size 64424509440 bytes
Drive TotalCylinders 61440
Drive TotalSectors 125829120
Drive TotalTracks 3932160
Drive TracksPerCylinder 64

Drive S:
Description Local Fixed Disk
Compressed Not Available
File System Not Available

APPENDIX C – TUNABLE PARAMETERS

Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #9, Partition #1
Partition Size 20.94 GB (22,480,420,864 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALDRIVE9
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 8
Drive SCSITargetId 0
Drive SectorsPerTrack 32
Drive Size 64424509440 bytes
Drive TotalCylinders 61440
Drive TotalSectors 125829120
Drive TotalTracks 3932160
Drive TracksPerCylinder 64

Drive T:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #11, Partition #1
Partition Size 26.37 GB (28,311,552,000 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALDRIVE11
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 9
Drive SCSITargetId 0
Drive SectorsPerTrack 32
Drive Size 436743438336 bytes
Drive TotalCylinders 416511
Drive TotalSectors 853014528
Drive TotalTracks 26656704
Drive TracksPerCylinder 64

Drive V:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 173.37 GB (186,157,871,104 bytes)
Free Space 72.28 GB (77,607,071,744 bytes)
Volume Name New Volume
Volume Serial Number 0424EB53
Partition Disk #4, Partition #0
Partition Size 173.37 GB (186,157,875,200 bytes)
Starting Offset 16384 bytes
Drive Description \\.\PHYSICALDRIVE4
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 1
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 4
Drive SCSITargetId 1
Drive SectorsPerTrack 32
Drive Size 186158940160 bytes
Drive TotalCylinders 177535
Drive TotalSectors 363591680
Drive TotalTracks 11362240
Drive TracksPerCylinder 64

Drive W:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 173.37 GB (186,157,871,104 bytes)
Free Space 21.76 GB (23,368,445,952 bytes)
Volume Name New Volume
Volume Serial Number 7CF245AC
Partition Disk #2, Partition #0
Partition Size 173.37 GB (186,157,875,200 bytes)
Starting Offset 16384 bytes
Drive Description \\.\PHYSICALDRIVE2
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 1
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 3
Drive SCSITargetId 1
Drive SectorsPerTrack 32
Drive Size 186158940160 bytes
Drive TotalCylinders 177535
Drive TotalSectors 363591680
Drive TotalTracks 11362240
Drive TracksPerCylinder 64

APPENDIX C – TUNABLE PARAMETERS

Drive X:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 173.37 GB (186,157,871,104 bytes)
Free Space 72.28 GB (77,607,096,320 bytes)
Volume Name New Volume
Volume Serial Number 4CDF2CAA
Partition Disk #8, Partition #0
Partition Size 173.37 GB (186,157,875,200 bytes)
Starting Offset 16384 bytes
Drive Description \\.\PHYSICALDRIVE8
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 1
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 7
Drive SCSITargetId 1
Drive SectorsPerTrack 32
Drive Size 186158940160 bytes
Drive TotalCylinders 177535
Drive TotalSectors 363591680
Drive TotalTracks 11362240
Drive TracksPerCylinder 64

Drive Y:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 173.37 GB (186,157,871,104 bytes)
Free Space 21.76 GB (23,368,491,008 bytes)
Volume Name New Volume
Volume Serial Number 586D1BB4
Partition Disk #10, Partition #0
Partition Size 173.37 GB (186,157,875,200 bytes)
Starting Offset 16384 bytes
Drive Description \\.\PHYSICALDRIVE10
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 1
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 8
Drive SCSITargetId 1
Drive SectorsPerTrack 32
Drive Size 186158940160 bytes
Drive TotalCylinders 177535
Drive TotalSectors 363591680
Drive TotalTracks 11362240
Drive TracksPerCylinder 64

Drive M:
Description Network Connection
Provider Name \\nrddata\g$

[SCSI]

Item Value
Name Hewlett Packard NetRAID-4M RAID Controller
Caption Hewlett Packard NetRAID-4M RAID Controller
Driver hpnsa
Status OK
PNP Device ID PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&267A616A&0&20
Device ID PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&267A616A&0&20
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 61
I/O Port 0x1000-0x10FF
Driver Not Available

Name Symbios Logic 896, 22910 PCI SCSI Adapter
Caption Symbios Logic 896, 22910 PCI SCSI Adapter
Driver sym_hi
Status OK
PNP Device ID PCI\VEN_1000&DEV_000B&SUBSYS_10001000&REV_05\3&267A616A&0&50
Device ID PCI\VEN_1000&DEV_000B&SUBSYS_10001000&REV_05\3&267A616A&0&50
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 58
I/O Port 0x1400-0x14FF
Driver c:\winnt\system32\drivers\sym_hi.sys (21136, 5.00.2134.1)

Name Symbios Logic 896, 22910 PCI SCSI Adapter
Caption Symbios Logic 896, 22910 PCI SCSI Adapter
Driver sym_hi
Status OK
PNP Device ID PCI\VEN_1000&DEV_000B&SUBSYS_10001000&REV_05\3&267A616A&0&51
Device ID PCI\VEN_1000&DEV_000B&SUBSYS_10001000&REV_05\3&267A616A&0&51
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 18
I/O Port 0x1800-0x18FF
Driver c:\winnt\system32\drivers\sym_hi.sys (21136, 5.00.2134.1)

Name Hewlett Packard NetRAID-4M RAID Controller
Caption Hewlett Packard NetRAID-4M RAID Controller
Driver hpnsa
Status OK
PNP Device ID PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&13C0B0C5&0&28
Device ID PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&13C0B0C5&0&28
Device Map Not Available

APPENDIX C – TUNABLE PARAMETERS

Index Not Available
Max Number Controlled Not Available
IRQ Number 44
I/O Port 0x2000-0x24FF
Driver Not Available

Name Hewlett Packard NetRAID-4M RAID Controller
Caption Hewlett Packard NetRAID-4M RAID Controller
Driver hpnsa
Status OK
PNP Device ID PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&13C0B0C5&0&30
Device ID PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&13C0B0C5&0&30
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 40
I/O Port 0x2400-0x24FF
Driver Not Available

Name Hewlett Packard NetRAID-4M RAID Controller
Caption Hewlett Packard NetRAID-4M RAID Controller
Driver hpnsa
Status OK
PNP Device ID PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&1070020&0&20
Device ID PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&1070020&0&20
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 32
I/O Port 0x3000-0x34FF
Driver Not Available

Name Hewlett Packard NetRAID-4M RAID Controller
Caption Hewlett Packard NetRAID-4M RAID Controller
Driver hpnsa
Status OK
PNP Device ID PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&1070020&0&28
Device ID PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&1070020&0&28
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 28
I/O Port 0x3400-0x34FF
Driver Not Available

Name Hewlett Packard NetRAID-4M RAID Controller
Caption Hewlett Packard NetRAID-4M RAID Controller
Driver hpnsa
Status OK
PNP Device ID PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&29E81982&0&20
Device ID PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&29E81982&0&20
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 24
I/O Port 0x4000-0x44FF
Driver Not Available

Name Hewlett Packard NetRAID-4M RAID Controller
Caption Hewlett Packard NetRAID-4M RAID Controller
Driver hpnsa
Status OK
PNP Device ID PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&29E81982&0&28
Device ID PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&29E81982&0&28
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 20
I/O Port 0x4400-0x44FF
Driver Not Available

[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
Intel 82371AB/EB PCI to USB Universal Host Controller
PCI\VEN_8086&DEV_7112&SUBSYS_00000000&REV_01\3&267A616A&0&7A
USB Root Hub USB\ROOT_HUB\4&B5B4E1B&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	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False	False
afd	AFD Networking Support Environment	c:\winnt\system32\drivers\afd.sys	Kernel Driver	True	Auto	Running	OK	Normal	False	True
aha154x	Aha154x	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False	False

APPENDIX C – TUNABLE PARAMETERS

aic116x	aic116x	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
aic78u2	aic78u2	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
aic78xx	aic78xx	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
ami0nt	ami0nt	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
amsint	amsint	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
asc	asc	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
asc3350p	asc3350p	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
asc3550	asc3550	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
asyncmac	RAS Asynchronous Media Driver			c:\winnt\system32\drivers\asyncmac.sys			Kernel Driver	False	
	Manual	Stopped	OK	Normal	False	False			
atapi	Standard IDE/ESDI Hard Disk Controller			c:\winnt\system32\drivers\atapi.sys			Kernel Driver	True	Boot
	Running	OK	Normal	False	True				
atdisk	Atdisk	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Ignore	False
	False								
atmarpc	ATM ARP Client Protocol		c:\winnt\system32\drivers\atmarpc.sys			Kernel Driver	False	Manual	
	Stopped	OK	Normal	False	False				
audstub	Audio Stub Driver			c:\winnt\system32\drivers\audstub.sys			Kernel Driver	True	Manual
	Running	OK	Normal	False	True				
beep	Beep	c:\winnt\system32\drivers\beep.sys			Kernel Driver	True	System	Running	OK
	Normal	False	True						
buslogic	BusLogic	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
cd20xmt	cd20xmt	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								
cirrus	cirrus	c:\winnt\system32\drivers\cirrus.sys			Kernel Driver	True	Manual	Running	OK
	Ignore	False	True						
cpqarray	Cpqarray	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
cpqarry2	cpqarry2	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
cpqfcalm	cpqfcalm	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
cpqfws2e	cpqfws2e	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
dac960nt	dac960nt	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
deckzpsx	deckzpsx	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
dfsdriver	DfsDriver	c:\winnt\system32\drivers\dfs.sys			File System Driver		True	Boot	
	Running	OK	Normal	False	True				
disk	Disk Driver	c:\winnt\system32\drivers\disk.sys			Kernel Driver	True	Boot	Running	OK
	Normal	False	True						
diskperf	Diskperf	c:\winnt\system32\drivers\diskperf.sys			Kernel Driver	True	Boot	Running	OK
	Normal	False	True						
dmboot	dmboot	c:\winnt\system32\drivers\dmboot.sys			Kernel Driver	False	Disabled	Stopped	OK
	Normal	False	False						
dmio	Logical Disk Manager Driver			c:\winnt\system32\drivers\dmio.sys			Kernel Driver	True	Boot
	Running	OK	Normal	False	True				
dmload	dmload	c:\winnt\system32\drivers\dmload.sys			Kernel Driver	True	Boot	Running	OK
	Normal	False	True						
efs	EFS	c:\winnt\system32\drivers\efs.sys			File System Driver		True	Disabled	
	Running	OK	Normal	False	True				
em	em	c:\winnt\system32\drivers\em.sys			Kernel Driver	True	System	Running	OK
	Normal	False	True						
fastfat	Fastfat	c:\winnt\system32\drivers\fastfat.sys			File System Driver		True	Disabled	
	Running	OK	Normal	False	True				
fd16_700	Fd16_700	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
fdc	Floppy Disk Controller Driver			c:\winnt\system32\drivers\fdc.sys			Kernel Driver	True	
	Manual	Running	OK	Normal	False	True			
fireport	fireport	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
flashpnt	flashpnt	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
flpydisk	Floppy Disk Driver		c:\winnt\system32\drivers\flpydisk.sys			Kernel Driver	True	Manual	
	Running	OK	Normal	False	True				
ftdisk	Volume Manager Driver		c:\winnt\system32\drivers\ftdisk.sys			Kernel Driver	True	Boot	
	Running	OK	Normal	False	True				
gpc	Generic Packet Classifier		c:\winnt\system32\drivers\msgpc.sys			Kernel Driver	True	Manual	
	Running	OK	Normal	False	True				
hpndisk	Hewlett Packard NetRAID-4M Disk Driver					c:\winnt\system32\drivers\hpndisk.sys		Kernel Driver	True
	Boot	Running	OK	Normal	False	True			
hpnmgt	HPNMgt	c:\winnt\system32\drivers\hpnmgt.sys			Kernel Driver	True	Boot	Running	OK
	Normal	False	True						
hpnsa	Hewlett Packard NetRAID-4M Driver			c:\winnt\system32\drivers\hpnsa.sys			Kernel Driver	True	Boot
	Running	OK	Normal	False	True				
hptx	HP 10/100TX PCI LAN Adapter NT Driver					c:\winnt\system32\drivers\hptxnt5.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	Intellde	c:\winnt\system32\drivers\intelide.sys			Kernel Driver	True	Boot	Running	OK
	Normal	False	True						
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	False	Manual	Stopped	OK
	Normal	False	False						
ipsraidn	ipsraidn	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
isapnp	PnP ISA/EISA Bus Driver		c:\winnt\system32\drivers\isapnp.sys			Kernel Driver	True	Boot	
	Running	OK	Critical	False	True				

APPENDIX C – TUNABLE PARAMETERS

kbdclass	Keyboard Class Driver	c:\winnt\system32\drivers\kbdclass.sys	Kernel Driver	True	System				
	Running	OK	Normal	False	True				
ksecdd	KSecDD	c:\winnt\system32\drivers\ksecdd.sys	Kernel Driver	True	Boot	Running	OK		
	Normal	False	True						
lbrtfdc	lbrtfdc	Not Available	Kernel Driver	False	System	Stopped	OK	Ignore	False
	False								
lp6nds35	lp6nds35	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
mnmdd	mnmdd	c:\winnt\system32\drivers\mnmdd.sys	Kernel Driver	True	System	Running	OK		
	Ignore	False	True						
modem	Modem	c:\winnt\system32\drivers\modem.sys	Kernel Driver	False	Manual	Stopped	OK		
	Ignore	False	False						
mouclass	Mouse Class Driver	c:\winnt\system32\drivers\mouclass.sys	Kernel Driver	True	System				
	Running	OK	Normal	False	True				
mountmgr	MountMgr	c:\winnt\system32\drivers\mountmgr.sys	Kernel Driver	True	Boot	Running	OK		
	Normal	False	True						
mraid35x	mraid35x	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
mrxsmb	MRXSMB	c:\winnt\system32\drivers\mrxsmb.sys	File System Driver	True	System				
	Running	OK	Normal	False	True				
msfs	Msfs	c:\winnt\system32\drivers\msfs.sys	File System Driver	True	System				
	Running	OK	Normal	False	True				
mskssrv	Microsoft Streaming Service Proxy	c:\winnt\system32\drivers\mskssrv.sys	Kernel Driver	False					
	Manual	Stopped	OK	Normal	False	False			
mspclock	Microsoft Streaming Clock Proxy	c:\winnt\system32\drivers\mspclock.sys	Kernel Driver	False					
	Manual	Stopped	OK	Normal	False	False			
mspqm	Microsoft Streaming Quality Manager Proxy	c:\winnt\system32\drivers\mspqm.sys	Kernel Driver	False					
	Manual	Stopped	OK	Normal	False	False			
mup	Mup	c:\winnt\system32\drivers\mup.sys	File System Driver	True	Boot				
	Running	OK	Normal	False	True				
ncrc710	Ncrc710	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
ndis	NDIS System Driver	c:\winnt\system32\drivers\ndis.sys	Kernel Driver	True	Boot				
	Running	OK	Normal	False	True				
ndistapi	Remote Access NDIS TAPI Driver	c:\winnt\system32\drivers\ndistapi.sys	Kernel Driver	True					
	Manual	Running	OK	Normal	False	True			
ndiswan	Remote Access NDIS WAN Driver	c:\winnt\system32\drivers\ndiswan.sys	Kernel Driver	True					
	Manual	Running	OK	Normal	False	True			
ndproxy	NDIS Proxy	c:\winnt\system32\drivers\ndproxy.sys	Kernel Driver	True	Manual	Running	OK		
	Normal	False	True						
netbios	NetBIOS Interface	c:\winnt\system32\drivers\netbios.sys	File System Driver	True					
	System	Running	OK	Normal	False	True			
netbt	NetBios over Tcpip	c:\winnt\system32\drivers\netbt.sys	Kernel Driver	True	System				
	Running	OK	Normal	False	True				
netdetect	NetDetect	c:\winnt\system32\drivers\netdtect.sys	Kernel Driver	False	Manual	Stopped	OK		
	Normal	False	False						
nm	Network Monitor Driver	c:\winnt\system32\drivers\nmnt.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			
parallel	Parallel	c:\winnt\system32\drivers\parallel.sys	Kernel Driver	False	Auto	Stopped	OK		
	Ignore	False	False						
parport	Parport	c:\winnt\system32\drivers\parport.sys	Kernel Driver	False	Auto	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	Auto	Stopped	OK		
	Ignore	False	False						
pci	PCI Bus Driver	c:\winnt\system32\drivers\pci.sys	Kernel Driver	True	Boot				
	Running	OK	Critical	False	True				
pcidump	PCIDump	Not Available	Kernel Driver	False	System	Stopped	OK	Ignore	False
	False								
pciide	PCIIde	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
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				
ql1080	ql1080	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
ql10wnt	Ql10wnt	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
ql1240	ql1240	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
ql2100	ql2100	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
rasacd	Remote Access Auto Connection Driver	c:\winnt\system32\drivers\rasacd.sys	Kernel Driver	True					
	System	Running	OK	Normal	False	True			
rasl2tp	WAN Miniport (L2TP)	c:\winnt\system32\drivers\rasl2tp.sys	Kernel Driver	True	Manual				
	Running	OK	Normal	False	True				
raspti	Direct Parallel	c:\winnt\system32\drivers\raspti.sys	Kernel Driver	True	Manual				
	Running	OK	Normal	False	True				
rca	Microsoft Streaming Network Raw Channel Access	c:\winnt\system32\drivers\rca.sys	Kernel Driver	False					
	Manual	Stopped	OK	Normal	False	False			
rdbss	Rdbss	c:\winnt\system32\drivers\rdbss.sys	File System Driver	True	System				
	Running	OK	Normal	False	True				
rdpwd	RDPWD	c:\winnt\system32\drivers\rdpwd.sys	Kernel Driver	False	Manual	Stopped	OK		
	Ignore	False	False						
redbook	Digital CD Audio Playback Filter Driver	c:\winnt\system32\drivers\redbook.sys	Kernel Driver	False					
	System	Stopped	OK	Normal	False	False			
serenum	Serenum Filter Driver	c:\winnt\system32\drivers\serenum.sys	Kernel Driver	True	Manual				
	Running	OK	Normal	False	True				
serial	Serial port driver	c:\winnt\system32\drivers\serial.sys	Kernel Driver	True	System				
	Running	OK	Ignore	False	True				

APPENDIX C – TUNABLE PARAMETERS

sfloppy	Sfloppy	c:\winnt\system32\drivers\sfloppy.sys	Kernel Driver	False	System	Stopped	OK	
	Ignore	False	False					
sglfb	sglfb	Not Available	Kernel Driver	False	System	Stopped	OK	Normal	False
	False							
simbad	Simbad	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False							
sparrow	Sparrow	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False							
spud	Special Purpose Utility Driver		c:\winnt\system32\drivers\spud.sys		Kernel Driver	True		
	Manual	Running	OK	Normal	False	True		
srv	Srv	c:\winnt\system32\drivers\srv.sys		File System Driver		True	Manual	
	Running	OK	Normal	False	True			
swenum	Software Bus Driver		c:\winnt\system32\drivers\swenum.sys		Kernel Driver	True	Manual	
	Running	OK	Normal	False	True			
symc810	symc810	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False							
symc8xx	symc8xx	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False							
sym_hi	sym_hi	c:\winnt\system32\drivers\sym_hi.sys		Kernel Driver	True	Boot	Running	OK
	Normal	False	True					
tcpip	TCP/IP Protocol Driver		c:\winnt\system32\drivers\tcpip.sys		Kernel Driver	True	System	
	Running	OK	Normal	False	True			
tdasync	TDASYNC	c:\winnt\system32\drivers\tdasync.sys		Kernel Driver	False	Manual	Stopped	OK
	Ignore	False	False					
tdipx	TDIPX	c:\winnt\system32\drivers\tdipx.sys		Kernel Driver	False	Manual	Stopped	OK
	Ignore	False	False					
tdnetb	TDNETB	c:\winnt\system32\drivers\tdnetb.sys		Kernel Driver	False	Manual	Stopped	OK
	Ignore	False	False					
tdpipe	TDPIPE	c:\winnt\system32\drivers\tdpipe.sys		Kernel Driver	False	Manual	Stopped	OK
	Ignore	False	False					
tdspx	TDSPX	c:\winnt\system32\drivers\tdspx.sys		Kernel Driver	False	Manual	Stopped	OK
	Ignore	False	False					
tdtcp	TDTCP	c:\winnt\system32\drivers\tdtcp.sys		Kernel Driver	False	Manual	Stopped	OK
	Ignore	False	False					
termdd	Terminal Device Driver		c:\winnt\system32\drivers\termdd.sys		Kernel Driver	False	Disabled	
	Stopped	OK	Normal	False	False			
tga	tga	Not Available	Kernel Driver	False	System	Stopped	OK	Ignore	False
	False							
udfs	Udfs	c:\winnt\system32\drivers\udfs.sys		File System Driver		False	Disabled	
	Stopped	OK	Normal	False	False			
uhcd	Microsoft USB Universal Host Controller Driver		c:\winnt\system32\drivers\uhcd.sys			Kernel Driver	True	
	Manual	Running	OK	Normal	False	True		
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	False	System	Stopped	OK
	Ignore	False	False					
wanarp	Remote Access IP ARP Driver		c:\winnt\system32\drivers\wanarp.sys		Kernel Driver	True		
	Manual	Running	OK	Normal	False	True		
wdica	WDICA	Not Available	Kernel Driver	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	C:\mksnt;C:\WINNT\system32;C:\WINNT;C:\WINNT\System32\Wbem;C:\PROGRA~1\MICROS~3\80\Tools\BINN;C:\PROGRA~1\MICROS~2\Common\MSDev98\Bin;C:\PROGRA~1\MICROS~2\Common\Tools;C:\PROGRA~1\MICROS~2\VC98\bin;	<SYSTEM>						
windir	%SystemRoot%	<SYSTEM>						
OS	Windows_NT	<SYSTEM>						
PROCESSOR_ARCHITECTURE	x86	<SYSTEM>						
PROCESSOR_LEVEL	6	<SYSTEM>						
PROCESSOR_IDENTIFIER	x86 Family 6 Model 10 Stepping 0, GenuineIntel	<SYSTEM>						
PROCESSOR_REVISION	0a00	<SYSTEM>						
NUMBER_OF_PROCESSORS	8	<SYSTEM>						
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH	<SYSTEM>						
TEMP	%SystemRoot%\TEMP	<SYSTEM>						
TMP	%SystemRoot%\TEMP	<SYSTEM>						
ROOTDIR	C:/	<SYSTEM>						
SHELL	C:/mksnt/sh.exe	<SYSTEM>						
HOME	C:/	<SYSTEM>						
TMPDIR	C:/WINNT/TEMP	<SYSTEM>						
TEMP	%USERPROFILE%\Local Settings\Temp	PRF_SUT6\Administrator						
TMP	%USERPROFILE%\Local Settings\Temp	PRF_SUT6\Administrator						
path	C:\Program Files\Microsoft Visual Studio\Common\MSDev98\Bin;C:\Program Files\Microsoft Visual Studio\Common\Tools;C:\Program Files\Microsoft Visual Studio\VC98\bin	PRF_SUT6\Administrator						
MSDevDir	C:\Program Files\Microsoft Visual Studio\Common\MSDev98	PRF_SUT6\Administrator						
include	C:\Program Files\Microsoft Visual Studio\VC98\atl\include;C:\Program Files\Microsoft Visual Studio\VC98\mfc\include;C:\Program Files\Microsoft Visual Studio\VC98\include	PRF_SUT6\Administrator						
lib	C:\Program Files\Microsoft Visual Studio\VC98\mfc\lib;C:\Program Files\Microsoft Visual Studio\VC98\lib	PRF_SUT6\Administrator						

[Jobs]

[Following are sub-categories of this main category]

[Print]

Document	Size	Owner	Notify	Status	Time Submitted		Start Time	Until Time	
	Elapsed Time		Pages Printed		Job ID	Priority	Parameters	Driver Name	Print
Processor	Host Print Queue		Data Type	Name					
Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	
	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	
	Unknown								

[Network Connections]

Local Name	Remote Name	Type	Status	User Name
M:	\\nrddata\g$	Disk	OK	PRF_SUT6\Administrator

[Running Tasks]

Name	Path	Process ID	Priority	Min Working Set	Max Working Set	Start Time
	Version	Size	File Date			
system idle process	Not Available	0	0	Not Available	Not Available	Not Available	Unknown
	Unknown	Unknown					

APPENDIX C – TUNABLE PARAMETERS

system Not Available 8 8 0 1413120 Not Available Unknown Unknown Unknown
smss.exe c:\winnt\system32\smss.exe 276 11 204800 1413120 12/18/2000 4:29:45 PM 5.00.2170.1 44.27 KB (45,328 bytes) 12/7/1999 4:00:00 AM
csrss.exe Not Available 304 13 Not Available Not Available 12/18/2000 4:29:53 PM Unknown Unknown Unknown
winlogon.exe c:\winnt\system32\winlogon.exe 324 13 204800 1413120 12/18/2000 4:29:55 PM 5.00.2182.1 173.27 KB (177,424 bytes) 12/7/1999 4:00:00 AM
services.exe c:\winnt\system32\services.exe 352 9 204800 1413120 12/18/2000 4:29:57 PM 5.00.2134.1 86.77 KB (88,848 bytes) 12/7/1999 4:00:00 AM
lsass.exe c:\winnt\system32\lsass.exe 364 13 204800 1413120 12/18/2000 4:29:57 PM 5.00.2184.1 32.77 KB (33,552 bytes) 12/7/1999 4:00:00 AM
svchost.exe c:\winnt\system32\svchost.exe 500 8 204800 1413120 12/18/2000 4:30:00 PM 5.00.2134.1 7.77 KB (7,952 bytes) 12/7/1999 4:00:00 AM
msdtc.exe c:\winnt\system32\msdtc.exe 528 8 204800 1413120 12/18/2000 4:30:01 PM 1999.9.3421.3 6.77 KB (6,928 bytes) 10/27/2000 9:37:51 AM
svchost.exe c:\winnt\system32\svchost.exe 776 8 204800 1413120 12/18/2000 4:30:04 PM 5.00.2134.1 7.77 KB (7,952 bytes) 12/7/1999 4:00:00 AM
afaagent.exe c:\program files\hp_raid\hpn\afaagent.exe 792 8 204800 1413120 12/18/2000 4:30:04 PM 2.4.0.4576 256.27 KB (262,416 bytes) 12/13/2000 3:08:09 PM
tcpsvcs.exe c:\winnt\system32\tcpsvcs.exe 856 8 204800 1413120 12/18/2000 4:30:05 PM 5.00.2134.1 24.77 KB (25,360 bytes) 12/7/1999 4:00:00 AM
snmp.exe c:\winnt\system32\snmp.exe 924 8 204800 1413120 12/18/2000 4:30:06 PM 5.00.2173.1 29.77 KB (30,480 bytes) 10/27/2000 9:37:33 AM
winmgmt.exe c:\winnt\system32\wbem\winmgmt.exe 940 8 204800 1413120 12/18/2000 4:30:06 PM 1.50.1085.0001 188.05 KB (192,567 bytes) 12/7/1999 4:00:00 AM
wins.exe c:\winnt\system32\wins.exe 1020 8 204800 1413120 12/18/2000 4:30:06 PM 5.00.2181.1 147.77 KB (151,312 bytes) 10/27/2000 9:37:42 AM
dns.exe c:\winnt\system32\dns.exe 1032 8 204800 1413120 12/18/2000 4:30:06 PM 5.00.2190.1 312.27 KB (319,760 bytes) 10/27/2000 9:37:43 AM
inetinfo.exe c:\winnt\system32\inetsrv\inetinfo.exe 1116 8 204800 1413120 12/18/2000 4:30:10 PM 5.00.0984 14.27 KB (14,608 bytes) 10/27/2000 9:38:36 AM
mssearch.exe c:\program files\common files\system\mssearch\bin\mssearch.exe 1260 8 204800 1413120 12/18/2000 4:30:10 PM 9.107.5512.0 72.00 KB (73,728 bytes) 10/30/2000 10:29:16 AM
svchost.exe c:\winnt\system32\svchost.exe 1592 8 204800 1413120 12/18/2000 4:30:47 PM 5.00.2134.1 7.77 KB (7,952 bytes) 12/7/1999 4:00:00 AM
locator.exe c:\winnt\system32\locator.exe 1692 8 204800 1413120 12/18/2000 4:32:15 PM 5.00.2135.1 70.77 KB (72,464 bytes) 12/7/1999 4:00:00 AM
explorer.exe c:\winnt\explorer.exe 1716 8 204800 1413120 12/18/2000 4:32:58 PM 5.00.2920.0000 232.77 KB (238,352 bytes) 12/7/1999 4:00:00 AM
sqlmangr.exe c:\program files\microsoft sql server\80\tools\binn\sqlmangr.exe 1464 8 204800 1413120 12/18/2000 4:33:01 PM 2000.080.0194.00 68.00 KB (69,632 bytes) 11/10/2000 11:45:54 AM
cmd.exe c:\winnt\system32\cmd.exe 1676 8 204800 1413120 12/18/2000 4:33:04 PM 5.00.2144.1 230.77 KB (236,304 bytes) 12/7/1999 4:00:00 AM
cmd.exe c:\winnt\system32\cmd.exe 1648 8 204800 1413120 12/18/2000 4:48:02 PM 5.00.2144.1 230.77 KB (236,304 bytes) 12/7/1999 4:00:00 AM
mmc.exe c:\winnt\system32\mmc.exe 1792 8 204800 1413120 12/19/2000 6:18:27 PM 5.00.2153.1 589.27 KB (603,408 bytes) 12/7/1999 4:00:00 AM
rsvp.exe c:\winnt\system32\rsvp.exe 1844 8 204800 1413120 12/19/2000 6:19:45 PM 5.00.2167.1 172.77 KB (176,912 bytes) 12/7/1999 4:00:00 AM

[Loaded Modules]

Name Version Size File Date Manufacturer Path

traffic.dll 5.00.2139.1 30.77 KB (31,504 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\traffic.dll
rsvp.exe 5.00.2167.1 172.77 KB (176,912 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\rsvp.exe
wbemprox.dll 1.50.1085.0001 40.05 KB (41,016 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\wbemprox.dll
rassapi.dll 5.00.2188.1 14.27 KB (14,608 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\rassapi.dll
adsnt.dll 5.00.2191.1 194.27 KB (198,928 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\adsnt.dll
dbghelp.dll 5.00.2195.1 159.27 KB (163,088 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\dbghelp.dll
localsec.dll 5.00.2134.1 227.27 KB (232,720 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\localsec.dll
devmgr.dll 5.00.2166.1 215.77 KB (220,944 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\devmgr.dll
filemgmt.dll 5.00.2134.1 287.27 KB (294,160 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\filemgmt.dll
pdh.dll 5.00.2174.1 143.27 KB (146,704 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\pdh.dll
smlogcfg.dll 5.00.2163.1 273.27 KB (279,824 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\smlogcfg.dll
cabinet.dll 5.00.2147.1 54.77 KB (56,080 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\cabinet.dll
msinfo32.dll 5.00.2177.1 312.27 KB (319,760 bytes) 10/27/2000 4:43:47 PM Microsoft Corporation c:\program files\common files\microsoft shared\msinfo\msinfo32.dll
riched20.dll 5.30.23.1200 421.27 KB (431,376 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\riched20.dll
riched32.dll 5.00.2134.1 3.77 KB (3,856 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\riched32.dll
els.dll 5.00.2175.1 151.27 KB (154,896 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\els.dll
ntmsmgr.dll 1,0,0,1 427.77 KB (438,032 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation and HighGround Systems, Inc. c:\winnt\system32\ntmsmgr.dll
mmfutil.dll 1.50.1085.0000 32.06 KB (32,829 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\mmfutil.dll
logdrive.dll 1.50.1085.0000 200.06 KB (204,863 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\logdrive.dll
dfrgres.dll 5.00.2150.1 27.50 KB (28,160 bytes) 12/7/1999 4:00:00 AM Executive Software International, Inc. c:\winnt\system32\dfrgres.dll
dfrgsnap.dll 5.00.2150.1 41.77 KB (42,768 bytes) 12/7/1999 4:00:00 AM Executive Software International, Inc. c:\winnt\system32\dfrgsnap.dll
dmdskres.dll 2191.1.296.2 119.00 KB (121,856 bytes) 12/7/1999 4:00:00 AM Microsoft Corp., VERITAS Software c:\winnt\system32\dmdskres.dll
dmutil.dll 2191.1.296.2 41.77 KB (42,768 bytes) 12/7/1999 4:00:00 AM VERITAS Software Corp. c:\winnt\system32\dmutil.dll
ntmsapi.dll 5.00.1948.1 50.27 KB (51,472 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\ntmsapi.dll
dmdskmgr.dll 2191.1.296.2 158.77 KB (162,576 bytes) 12/7/1999 4:00:00 AM Microsoft Corp., VERITAS Software c:\winnt\system32\dmdskmgr.dll
mycomput.dll 5.00.2134.1 107.77 KB (110,352 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\mycomput.dll
mmcndmgr.dll 5.00.2178.1 815.27 KB (834,832 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\mmcndmgr.dll
mfc42u.dll 6.00.8665.0 972.05 KB (995,384 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\mfc42u.dll

APPENDIX C – TUNABLE PARAMETERS

mmc.exe 5.00.2153.1 589.27 KB (603,408 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\mmc.exe
cmd.exe 5.00.2144.1 230.77 KB (236,304 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\cmd.exe
sqlmangr.rll 2000.080.0194.00 96.00 KB (98,304 bytes) 11/10/2000 11:45:54 AM Microsoft Corporation
c:\program files\microsoft sql server\80\tools\binn\resources\1033\sqlmangr.rll
sqlsvc.rll 2000.080.0194.00 24.00 KB (24,576 bytes) 11/10/2000 11:45:51 AM Microsoft Corporation
c:\program files\microsoft sql server\80\tools\binn\resources\1033\sqlsvc.rll
sqlresld.dll 2000.080.0194.00 28.06 KB (28,738 bytes) 11/10/2000 11:45:51 AM Microsoft Corporation
c:\program files\microsoft sql server\80\tools\binn\sqlresld.dll
odbcbcp.dll 2000.080.0194.00 28.07 KB (28,742 bytes) 10/30/2000 11:02:12 AM Microsoft Corporation
c:\winnt\system32\odbcbcp.dll
sqlsvc.dll 2000.080.0194.00 92.06 KB (94,272 bytes) 11/10/2000 11:45:51 AM Microsoft Corporation
c:\program files\microsoft sql server\80\tools\binn\sqlsvc.dll
sqlunirl.dll 2000.080.0194.00 176.06 KB (180,290 bytes) 8/6/2000 2:51:56 AM Microsoft Corporation
c:\winnt\system32\sqlunirl.dll
w95scm.dll 2000.080.0194.00 48.06 KB (49,216 bytes) 11/10/2000 11:45:51 AM Microsoft Corporation
c:\program files\microsoft sql server\80\tools\binn\w95scm.dll
sqlmangr.exe 2000.080.0194.00 68.00 KB (69,632 bytes) 11/10/2000 11:45:54 AM Microsoft Corporation c:\program files\microsoft sql server\80\tools\binn\sqlmangr.exe
imm32.dll 5.00.2180.1 93.77 KB (96,016 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\imm32.dll
actxprxy.dll 5.00.2920.0000 70.77 KB (72,464 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\actxprxy.dll
usp10.dll 1.0325.2180.1 307.77 KB (315,152 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\usp10.dll
faxshell.dll 5.00.2134.1 8.27 KB (8,464 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\faxshell.dll
msacm32.dll 5.00.2134.1 65.27 KB (66,832 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\msacm32.dll
avifil32.dll 5.00.2134.1 76.27 KB (78,096 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\avifil32.dll
msvfw32.dll 5.00.2134.1 113.77 KB (116,496 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\msvfw32.dll
docprop2.dll 5.00.2178.1 297.77 KB (304,912 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\docprop2.dll
webvw.dll 5.00.2920.0000 1.06 MB (1,115,408 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\webvw.dll
msls31.dll 3.10.337.0 145.27 KB (148,752 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\msls31.dll
shdoclc.dll 5.00.2920.0000 324.50 KB (332,288 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\shdoclc.dll
wininet.dll 5.00.2920.0000 456.77 KB (467,728 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\wininet.dll
mshtml.dll 5.00.2920.0000 2.25 MB (2,357,008 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\mshtml.dll
mlang.dll 5.00.2920.0000 510.77 KB (523,024 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\mlang.dll
urlmon.dll 5.00.2920.0000 426.77 KB (437,008 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\urlmon.dll
browselc.dll 5.00.2920.0000 34.50 KB (35,328 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\browselc.dll
linkinfo.dll 5.00.2134.1 15.77 KB (16,144 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\linkinfo.dll
powrprof.dll 5.00.2920.0000 13.27 KB (13,584 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\powrprof.dll
batmeter.dll 5.00.2920.0000 20.27 KB (20,752 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\batmeter.dll
stobject.dll 5.00.2144.1 81.77 KB (83,728 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\stobject.dll
msi.dll 1.10.1029.0 1.71 MB (1,794,320 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\msi.dll
webcheck.dll 5.00.2920.0000 251.77 KB (257,808 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\webcheck.dll
ntshrui.dll 5.00.2134.1 46.77 KB (47,888 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\ntshrui.dll
mydocs.dll 5.00.2920.0000 55.77 KB (57,104 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\mydocs.dll
browseui.dll 5.00.2920.0000 793.27 KB (812,304 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\browseui.dll
shdocvw.dll 5.00.2920.0000 1.05 MB (1,104,144 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\shdocvw.dll
explorer.exe 5.00.2920.0000 232.77 KB (238,352 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\explorer.exe
locator.exe 5.00.2135.1 70.77 KB (72,464 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\locator.exe
tapisrv.dll 5.00.2186.1 168.77 KB (172,816 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\tapisrv.dll
iprop.dll 5.00.2181.1 4.27 KB (4,368 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\iprop.dll
srchidx.dll 9.107.5512.0 433.50 KB (443,904 bytes) 10/30/2000 10:29:18 AM Microsoft Corporation
c:\progra~1\common~1\system\mssearch\bin\srchidx.dll
propdefs.dll 9.107.5512.0 164.00 KB (167,936 bytes) 10/30/2000 10:29:17 AM Microsoft Corporation
c:\progra~1\common~1\system\mssearch\bin\propdefs.dll
lcdetect.dll 9.107.5512.0 31.00 KB (31,744 bytes) 10/30/2000 10:29:16 AM Microsoft Corporation c:\program files\common files\system\mssearch\bin\lcdetect.dll
tquery.dll 9.107.5512.0 1.61 MB (1,690,112 bytes) 10/30/2000 10:29:18 AM Microsoft Corporation c:\program files\common files\system\mssearch\bin\tquery.dll
mssrch.dll 9.107.5512.0 1.49 MB (1,566,976 bytes) 10/30/2000 10:29:17 AM Microsoft Corporation
c:\progra~1\common~1\system\mssearch\bin\mssrch.dll
mssws.dll 9.107.5512.0 18.94 KB (19,392 bytes) 10/30/2000 10:29:17 AM Microsoft Corporation c:\program files\common files\system\mssearch\bin\mssws.dll
mssearch.exe 9.107.5512.0 72.00 KB (73,728 bytes) 10/30/2000 10:29:16 AM Microsoft Corporation
c:\program files\common files\system\mssearch\bin\mssearch.exe
ntfsdrv.dll 5.00.0984 36.77 KB (37,648 bytes) 10/27/2000 9:40:37 AM Microsoft Corporation
c:\winnt\system32\inetsrv\ntfsdrv.dll
aqueue.dll 5.00.0984 260.27 KB (266,512 bytes) 10/27/2000 9:40:36 AM Microsoft Corporation
c:\winnt\system32\inetsrv\aqueue.dll
seo.dll 5.00.0984 230.77 KB (236,304 bytes) 10/27/2000 9:40:37 AM Microsoft Corporation
c:\winnt\system32\inetsrv\seo.dll
iscomlog.dll 5.00.0984 24.77 KB (25,360 bytes) 10/27/2000 9:38:37 AM Microsoft Corporation
c:\winnt\system32\inetsrv\iscomlog.dll
lonsint.dll 5.00.0984 11.77 KB (12,048 bytes) 10/27/2000 9:38:37 AM Microsoft Corporation
c:\winnt\system32\inetsrv\lonsint.dll
inetsloc.dll 5.00.0984 20.27 KB (20,752 bytes) 10/27/2000 9:38:38 AM Microsoft Corporation
c:\winnt\system32\inetsloc.dll
staxmem.dll 5.00.0984 8.27 KB (8,464 bytes) 10/27/2000 9:38:36 AM Microsoft Corporation
c:\winnt\system32\staxmem.dll
exstrace.dll 5.00.0984 13.77 KB (14,096 bytes) 10/27/2000 9:38:38 AM Microsoft Corporation
c:\winnt\system32\exstrace.dll
rwnh.dll 5.00.0984 10.77 KB (11,024 bytes) 10/27/2000 9:40:37 AM Microsoft Corporation
c:\winnt\system32\rwnh.dll

APPENDIX C – TUNABLE PARAMETERS

fcachdll.dll 5.00.0984 43.77 KB (44,816 bytes) 10/27/2000 9:40:36 AM Microsoft Corporation
c:\winnt\system32\fcachdll.dll
iisfecnv.dll 5.00.0984 7.27 KB (7,440 bytes) 10/27/2000 9:38:36 AM Microsoft Corporation
c:\winnt\system32\inetsrv\iisfecnv.dll
isatq.dll 5.00.0984 61.27 KB (62,736 bytes) 10/27/2000 9:38:39 AM Microsoft Corporation
c:\winnt\system32\inetsrv\isatq.dll
infocomm.dll 5.00.0984 234.27 KB (239,888 bytes) 10/27/2000 9:38:37 AM Microsoft Corporation
c:\winnt\system32\inetsrv\infocomm.dll
smtpsvc.dll 5.00.0984 406.27 KB (416,016 bytes) 10/27/2000 9:40:39 AM Microsoft Corporation
c:\winnt\system32\inetsrv\smtpsvc.dll
security.dll 5.00.2154.1 5.77 KB (5,904 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\security.dll
svcext.dll 5.00.0984 39.77 KB (40,720 bytes) 10/27/2000 9:38:37 AM Microsoft Corporation
c:\winnt\system32\inetsrv\svcext.dll
admexs.dll 5.00.0984 27.77 KB (28,432 bytes) 10/27/2000 9:38:36 AM Microsoft Corporation
c:\winnt\system32\inetsrv\admexs.dll
wamreg.dll 5.00.0984 46.27 KB (47,376 bytes) 10/27/2000 9:38:45 AM Microsoft Corporation
c:\winnt\system32\inetsrv\wamreg.dll
metadata.dll 5.00.0984 70.77 KB (72,464 bytes) 10/27/2000 9:38:37 AM Microsoft Corporation
c:\winnt\system32\inetsrv\metadata.dll
iismap.dll 5.00.0984 56.27 KB (57,616 bytes) 10/27/2000 9:38:38 AM Microsoft Corporation
c:\winnt\system32\iismap.dll
nsepm.dll 5.00.0984 43.27 KB (44,304 bytes) 10/27/2000 9:38:37 AM Microsoft Corporation
c:\winnt\system32\inetsrv\nsepm.dll
admwprox.dll 5.00.0984 31.77 KB (32,528 bytes) 10/27/2000 9:38:38 AM Microsoft Corporation
c:\winnt\system32\admwprox.dll
coadmin.dll 5.00.0984 39.77 KB (40,720 bytes) 10/27/2000 9:38:39 AM Microsoft Corporation
c:\winnt\system32\inetsrv\coadmin.dll
iisadmin.dll 5.00.0984 14.77 KB (15,120 bytes) 10/27/2000 9:38:36 AM Microsoft Corporation
c:\winnt\system32\inetsrv\iisadmin.dll
rpcref.dll 5.00.0984 4.27 KB (4,368 bytes) 10/27/2000 9:38:37 AM Microsoft Corporation
c:\winnt\system32\inetsrv\rpcref.dll
inetinfo.exe 5.00.0984 14.27 KB (14,608 bytes) 10/27/2000 9:38:36 AM Microsoft Corporation
c:\winnt\system32\inetsrv\inetinfo.exe
dns.exe 5.00.2190.1 312.27 KB (319,760 bytes) 10/27/2000 9:37:43 AM Microsoft Corporation
c:\winnt\system32\dns.exe
wins.exe 5.00.2181.1 147.77 KB (151,312 bytes) 10/27/2000 9:37:42 AM Microsoft Corporation
c:\winnt\system32\wins.exe
wshnetbs.dll 5.00.2134.1 7.77 KB (7,952 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\wshnetbs.dll
rapilib.dll 5.00.2167.1 25.27 KB (25,872 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\rapilib.dll
rsvpsp.dll 5.00.2167.1 74.77 KB (76,560 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\rsvpsp.dll
ntmarta.dll 5.00.2158.1 98.77 KB (101,136 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\ntmarta.dll
provthrd.dll 1.50.1085.0000 68.07 KB (69,708 bytes) 10/27/2000 4:43:37 PM Microsoft Corporation
c:\winnt\system32\wbem\provthrd.dll
ntevt.dll 1.50.1085.0000 192.06 KB (196,669 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\ntevt.dll
psapi.dll 5.00.2134.1 28.27 KB (28,944 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\psapi.dll
framedyn.dll 1.50.1085.0000 164.05 KB (167,992 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\framedyn.dll
cimwin32.dll 1.50.1085.0000 1.03 MB (1,077,306 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\cimwin32.dll

wbemsvc.dll 1.50.1085.0000 140.07 KB (143,430 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\wbemsvc.dll
wbemess.dll 1.50.1085.0001 352.05 KB (360,503 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\wbemess.dll
fastprox.dll 1.50.1085.0001 144.08 KB (147,534 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\fastprox.dll
wbemcore.dll 1.50.1085.0001 632.05 KB (647,224 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\wbemcore.dll
wbemcomn.dll 1.50.1085.0001 684.05 KB (700,472 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\wbem\wbemcomn.dll
winmgmt.exe 1.50.1085.0001 188.05 KB (192,567 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\winmgmt.exe
odbcint.dll 3.520.6526.0 88.00 KB (90,112 bytes) 10/30/2000 11:02:08 AM Microsoft Corporation
c:\winnt\system32\odbcint.dll
odbc32.dll 3.520.6526.0 216.27 KB (221,456 bytes) 10/30/2000 11:02:07 AM Microsoft Corporation
c:\winnt\system32\odbc32.dll
sqlsnmp.dll 2000.080.0194.00 60.00 KB (61,440 bytes) 11/10/2000 11:45:06 AM Microsoft Corporation
c:\program files\microsoft sql server\mssql\binn\sqlsnmp.dll
ftpmib.dll 5.00.0984 6.27 KB (6,416 bytes) 10/27/2000 9:38:45 AM Microsoft Corporation
c:\winnt\system32\inetsrv\ftpmib.dll
iisrtl.dll 5.00.0984 120.77 KB (123,664 bytes) 10/27/2000 9:38:38 AM Microsoft Corporation
c:\winnt\system32\iisrtl.dll
infoadmn.dll 5.00.0984 12.27 KB (12,560 bytes) 10/27/2000 9:38:38 AM Microsoft Corporation
c:\winnt\system32\infoadmn.dll
httpmib.dll 5.00.0984 9.27 KB (9,488 bytes) 10/27/2000 9:38:42 AM Microsoft Corporation
c:\winnt\system32\inetsrv\httpmib.dll
loadperf.dll 5.00.2195.1 60.77 KB (62,224 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\loadperf.dll
iasperf.dll 5.00.2160.1 20.27 KB (20,752 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\iasperf.dll
dsauth.dll 5.00.2165.1 67.77 KB (69,392 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\dsauth.dll
dhcpsapi.dll 5.00.2165.1 88.77 KB (90,896 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\dhcpsapi.dll
dhcpmib.dll 5.00.2134.1 8.77 KB (8,976 bytes) 10/27/2000 4:43:10 PM Microsoft Corporation
c:\winnt\system32\dhcpmib.dll
winsrpc.dll 5.00.2134.1 13.27 KB (13,584 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\winsrpc.dll
winsmib.dll 5.00.2134.1 22.77 KB (23,312 bytes) 10/27/2000 9:37:42 AM Microsoft Corporation
c:\winnt\system32\winsmib.dll
rtipxmib.dll 5.00.2168.1 29.77 KB (30,480 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\rtipxmib.dll
perfos.dll 5.00.2155.1 21.27 KB (21,776 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\perfos.dll
btpagnt.dll 5.00.2168.1 13.27 KB (13,584 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\btpagnt.dll
ospfagnt.dll 5.00.2168.1 6.77 KB (6,928 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\ospfagnt.dll
ripagnt.dll 5.00.2168.1 24.27 KB (24,848 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\ripagnt.dll
mcastmib.dll 5.00.2168.1 13.27 KB (13,584 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\mcastmib.dll
igmpagnt.dll 5.00.2168.1 8.77 KB (8,976 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\igmpagnt.dll
acsmib.dll 5.00.2167.1 11.27 KB (11,536 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\acsmib.dll

APPENDIX C – TUNABLE PARAMETERS

evntagnt.dll 5.00.2167.1 94.77 KB (97,040 bytes) 10/27/2000 9:37:33 AM Microsoft Corporation c:\winnt\system32\evntagnt.dll
snmpmib.dll 5.00.2134.1 5.77 KB (5,904 bytes) 10/27/2000 9:37:33 AM Microsoft Corporation c:\winnt\system32\snmpmib.dll
inetmib1.dll 5.00.2168.1 28.77 KB (29,456 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\inetmib1.dll
lmmib2.dll 5.00.2134.1 29.27 KB (29,968 bytes) 10/27/2000 9:37:33 AM Microsoft Corporation c:\winnt\system32\lmmib2.dll
snmpapi.dll 5.00.2134.1 17.27 KB (17,680 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\snmpapi.dll
snmp.exe 5.00.2173.1 29.77 KB (30,480 bytes) 10/27/2000 9:37:33 AM Microsoft Corporation c:\winnt\system32\snmp.exe
simptcp.dll 5.00.2134.1 19.27 KB (19,728 bytes) 10/27/2000 9:37:44 AM Microsoft Corporation c:\winnt\system32\simptcp.dll
tcpsvcs.exe 5.00.2134.1 24.77 KB (25,360 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\tcpsvcs.exe
afapopup.dll 2.4.0.4576 40.00 KB (40,960 bytes) 12/13/2000 3:08:09 PM Adaptec, Inc. c:\program files\hp_raid\hpn\afapopup.dll
afaappse.dll 2.4.0.4576 176.00 KB (180,224 bytes) 12/13/2000 3:08:08 PM Adaptec, Inc. c:\program files\hp_raid\hpn\afaappse.dll
afaapi.dll 2.4.0.4576 856.27 KB (876,816 bytes) 12/13/2000 3:08:08 PM Adaptec, Inc. c:\program files\hp_raid\hpn\afaapi.dll
rpcns4.dll 5.00.2135.1 23.77 KB (24,336 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\rpcns4.dll
afaagent.exe 2.4.0.4576 256.27 KB (262,416 bytes) 12/13/2000 3:08:09 PM Adaptec, Inc. c:\program files\hp_raid\hpn\afaagent.exe
wmi.dll 5.00.2191.1 6.27 KB (6,416 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\wmi.dll
netshell.dll 5.00.2176.1 456.77 KB (467,728 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\netshell.dll
netman.dll 5.00.2175.1 88.77 KB (90,896 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\netman.dll
rasdlg.dll 5.00.2194.1 514.27 KB (526,608 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\rasdlg.dll
netcfgx.dll 5.00.2175.1 533.77 KB (546,576 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\netcfgx.dll
rasmans.dll 5.00.2188.1 146.77 KB (150,288 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\rasmans.dll
iashlpr.dll 5.00.2184.1 33.27 KB (34,064 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\iashlpr.dll
iasacct.dll 5.00.2134.1 28.27 KB (28,944 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\iasacct.dll
iasuserr.dll 5.00.2134.1 25.77 KB (26,384 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\iasuserr.dll
iasnap.dll 5.00.2134.1 58.77 KB (60,176 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\iasnap.dll
iaspipe.dll 5.00.2134.1 41.77 KB (42,768 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\iaspipe.dll
expsrv.dll 6.0.8540 370.27 KB (379,152 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\expsrv.dll
vbajet32.dll 6.1.8268 30.27 KB (30,992 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\vbajet32.dll
msjtes40.dll 4.00.2927.8 232.27 KB (237,840 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\msjtes40.dll
oledb32r.dll 2.60.6526.0 68.27 KB (69,904 bytes) 10/30/2000 11:02:08 AM Microsoft Corporation c:\program files\common files\system\ole db\oledb32r.dll

msdart.dll 2.60.6526.0 144.27 KB (147,728 bytes) 10/30/2000 11:02:06 AM Microsoft Corporation c:\winnt\system32\msdart.dll
oledb32.dll 2.60.6526.0 448.27 KB (459,024 bytes) 10/30/2000 11:02:08 AM Microsoft Corporation c:\program files\common files\system\ole db\oledb32.dll
msjint40.dll 4.00.2927.2 148.27 KB (151,824 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\msjint40.dll
msjter40.dll 4.00.2927.2 52.27 KB (53,520 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\msjter40.dll
mswstr10.dll 4.00.2927.10 600.27 KB (614,672 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\mswstr10.dll
msjet40.dll 4.00.2927.4 1.43 MB (1,495,312 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\msjet40.dll
msjetoledb40.dll 4.00.2927.2 340.27 KB (348,432 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\msjetoledb40.dll
iasrad.dll 5.00.2139.1 94.27 KB (96,528 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\iasrad.dll
iassam.dll 5.00.2160.1 96.27 KB (98,576 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\iassam.dll
iasads.dll 5.00.2134.1 73.77 KB (75,536 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\iasads.dll
iaspolcy.dll 5.00.2134.1 25.27 KB (25,872 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\iaspolcy.dll
iassvcs.dll 5.00.2160.1 58.77 KB (60,176 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\iassvcs.dll
iassdo.dll 5.00.2157.1 262.27 KB (268,560 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\iassdo.dll
sens.dll 5.00.2163.1 36.77 KB (37,648 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\sens.dll
ias.dll 5.00.2134.1 7.27 KB (7,440 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\ias.dll
es.dll 1999.9.3422.21 231.77 KB (237,328 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\es.dll
mtxoci.dll 1999.9.3421.3 109.27 KB (111,888 bytes) 10/27/2000 9:37:52 AM Microsoft Corporation c:\winnt\system32\mtxoci.dll
resutils.dll 5.00.2191.1 39.77 KB (40,720 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\resutils.dll
clusapi.dll 5.00.2179.1 50.27 KB (51,472 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\clusapi.dll
msvcp50.dll 5.00.7051 552.50 KB (565,760 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\msvcp50.dll
xolehlp.dll 1999.9.3421.3 17.27 KB (17,680 bytes) 10/27/2000 9:37:51 AM Microsoft Corporation c:\winnt\system32\xolehlp.dll
msdtclog.dll 1999.9.3421.3 89.77 KB (91,920 bytes) 10/27/2000 9:37:51 AM Microsoft Corporation c:\winnt\system32\msdtclog.dll
mtxclu.dll 1999.9.3421.3 50.27 KB (51,472 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\mtxclu.dll
msdtcprx.dll 1999.9.3422.10 619.27 KB (634,128 bytes) 10/27/2000 9:37:52 AM Microsoft Corporation c:\winnt\system32\msdtcprx.dll
txfaux.dll 1999.9.3422.24 341.27 KB (349,456 bytes) 10/27/2000 9:37:51 AM Microsoft Corporation c:\winnt\system32\txfaux.dll
msdtctm.dll 1999.9.3422.12 1.02 MB (1,070,864 bytes) 10/27/2000 9:37:51 AM Microsoft Corporation c:\winnt\system32\msdtctm.dll
msdtc.exe 1999.9.3421.3 6.77 KB (6,928 bytes) 10/27/2000 9:37:51 AM Microsoft Corporation c:\winnt\system32\msdtc.exe
winrnr.dll 5.00.2160.1 18.77 KB (19,216 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\winrnr.dll

APPENDIX C – TUNABLE PARAMETERS

rpcss.dll	5.00.2181.1	229.27 KB (234,768 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\rpcss.dll			
svchost.exe	5.00.2134.1	7.77 KB (7,952 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\svchost.exe			
scecli.dll	5.00.2191.1	105.27 KB (107,792 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\scecli.dll			
atl.dll	3.00.8449	57.56 KB (58,938 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\atl.dll			
certcli.dll	5.00.2175.1	132.27 KB (135,440 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\certcli.dll			
mswsock.dll	5.00.2152.1	62.27 KB (63,760 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\mswsock.dll			
ntdsatq.dll	5.00.2181.1	31.27 KB (32,016 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\ntdsatq.dll			
ntdsa.dll	5.00.2195.1	993.27 KB (1,017,104 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\ntdsa.dll			
kdcsvc.dll	5.00.2181.1	133.77 KB (136,976 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\kdcsvc.dll			
sfmapi.dll	5.00.2134.1	38.77 KB (39,696 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\sfmapi.dll			
rassfm.dll	5.00.2168.1	21.27 KB (21,776 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\rassfm.dll			
schannel.dll	5.00.2170.1	139.77 KB (143,120 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\schannel.dll			
netlogon.dll	5.00.2182.1	347.77 KB (356,112 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\netlogon.dll			
msv1_0.dll	5.00.2164.1	94.77 KB (97,040 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\msv1_0.dll			
kerberos.dll	5.00.2181.1	196.77 KB (201,488 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\kerberos.dll			
msprivs.dll	5.00.2154.1	41.50 KB (42,496 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\msprivs.dll			
samsrv.dll	5.00.2192.1	357.77 KB (366,352 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\samsrv.dll			
lsasrv.dll	5.00.2184.1	487.77 KB (499,472 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\lsasrv.dll			
lsass.exe	5.00.2184.1	32.77 KB (33,552 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\lsass.exe			
esent.dll	6.0.3939.6	1.07 MB (1,120,016 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\esent.dll			
rasadhlp.dll	5.00.2168.1	7.27 KB (7,440 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\rasadhlp.dll			
wshtcpip.dll	5.00.2134.1	17.27 KB (17,680 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\wshtcpip.dll			
msafd.dll	5.00.2153.1	54.27 KB (55,568 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\msafd.dll			
clbcatq.dll	1999.9.3422.14	479.27 KB (490,768 bytes)	10/27/2000 9:37:44 AM	Microsoft Corporation
	c:\winnt\system32\clbcatq.dll			
dhcpcsvc.dll	5.00.2153.1	88.77 KB (90,896 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\dhcpcsvc.dll			
tapi32.dll	5.00.2182.1	123.27 KB (126,224 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\tapi32.dll			
rasman.dll	5.00.2188.1	54.77 KB (56,080 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\rasman.dll			
rasapi32.dll	5.00.2188.1	189.77 KB (194,320 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\rasapi32.dll			
rtutils.dll	5.00.2168.1	43.77 KB (44,816 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\rtutils.dll			
adsldpc.dll	5.00.2172.1	127.77 KB (130,832 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\adsldpc.dll			
activeds.dll	5.00.2172.1	172.77 KB (176,912 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\activeds.dll			
oleaut32.dll	2.40.4512	600.27 KB (614,672 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\oleaut32.dll			
mprapi.dll	5.00.2181.1	79.27 KB (81,168 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\mprapi.dll			
iphlpapi.dll	5.00.2173.2	67.77 KB (69,392 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\iphlpapi.dll			
rnr20.dll	5.00.2152.1	35.77 KB (36,624 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\rnr20.dll			
wmicore.dll	5.00.2178.1	70.77 KB (72,464 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\wmicore.dll			
psbase.dll	5.00.2146.1	111.77 KB (114,448 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\psbase.dll			
cryptsvc.dll	5.00.2181.1	61.77 KB (63,248 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\cryptsvc.dll			
cryptdll.dll	5.00.2135.1	41.27 KB (42,256 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\cryptdll.dll			
wkssvc.dll	5.00.2181.1	95.27 KB (97,552 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\wkssvc.dll			
srvsvc.dll	5.00.2178.1	79.27 KB (81,168 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\srvsvc.dll			
cfgmgr32.dll	5.00.2134.1	16.77 KB (17,168 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\cfgmgr32.dll			
dmserver.dll	2191.1.296.2	11.77 KB (12,048 bytes)	12/7/1999 4:00:00 AM	VERITAS Software Corp.
	c:\winnt\system32\dmserver.dll			
winsta.dll	5.00.2134.1	36.27 KB (37,136 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\winsta.dll			
icmp.dll	5.00.2134.1	7.27 KB (7,440 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\icmp.dll			
lmhsvc.dll	5.00.2134.1	9.27 KB (9,488 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\lmhsvc.dll			
eventlog.dll	5.00.2178.1	43.77 KB (44,816 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\eventlog.dll			
ntdsapi.dll	5.00.2160.1	56.27 KB (57,616 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\ntdsapi.dll			
scesrv.dll	5.00.2188.1	225.77 KB (231,184 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\scesrv.dll			
umpnpmgr.dll	5.00.2182.1	86.27 KB (88,336 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\umpnpmgr.dll			
services.exe	5.00.2134.1	86.77 KB (88,848 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\services.exe			
netmsg.dll	5.00.2137.1	152.50 KB (156,160 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\netmsg.dll			
comdlg32.dll	5.00.2920.0000	236.77 KB (242,448 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\comdlg32.dll			
netui2.dll	5.00.2134.1	280.27 KB (286,992 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\netui2.dll			
mprui.dll	5.00.2134.1	54.77 KB (56,080 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\mprui.dll			
netui1.dll	5.00.2134.1	210.27 KB (215,312 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\netui1.dll			

APPENDIX C – TUNABLE PARAMETERS

netui0.dll	5.00.2134.1	70.27 KB (71,952 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\netui0.dll			
ntlanman.dll	5.00.2157.1	35.27 KB (36,112 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\ntlanman.dll			
mpr.dll	5.00.2146.1	53.27 KB (54,544 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\mpr.dll			
cscui.dll	5.00.2172.1	227.27 KB (232,720 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\cscui.dll			
winspool.drv	5.00.2167.1	109.77 KB (112,400 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\winspool.drv			
winscard.dll	5.00.2134.1	77.27 KB (79,120 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\winscard.dll			
wlnotify.dll	5.00.2164.1	53.27 KB (54,544 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\wlnotify.dll			
cscdll.dll	5.00.2189.1	98.27 KB (100,624 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\cscdll.dll			
lz32.dll	5.00.2134.1	9.77 KB (10,000 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\lz32.dll			
version.dll	5.00.2134.1	15.77 KB (16,144 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\version.dll			
rsabase.dll	5.00.2150.1	128.77 KB (131,856 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\rsabase.dll			
mscat32.dll	5.131.2134.1	7.77 KB (7,952 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\mscat32.dll			
ole32.dll	5.00.2181.1	966.27 KB (989,456 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\ole32.dll			
imagehlp.dll	5.00.2195.1	125.27 KB (128,272 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\imagehlp.dll			
msasn1.dll	5.00.2134.1	51.27 KB (52,496 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\msasn1.dll			
crypt32.dll	5.131.2173.1	465.77 KB (476,944 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\crypt32.dll			
wintrust.dll	5.131.2143.1	162.27 KB (166,160 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\wintrust.dll			
setupapi.dll	5.00.2183.1	554.27 KB (567,568 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\setupapi.dll			
winmm.dll	5.00.2161.1	184.77 KB (189,200 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\winmm.dll			
comctl32.dll	5.81	540.27 KB (553,232 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\comctl32.dll			
shlwapi.dll	5.00.2920.0000	282.77 KB (289,552 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\shlwapi.dll			
shell32.dll	5.00.2920.0000	2.24 MB (2,352,400 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\shell32.dll			
msgina.dll	5.00.2191.1	309.77 KB (317,200 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\msgina.dll			
wsock32.dll	5.00.2152.1	21.27 KB (21,776 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\wsock32.dll			
dnsapi.dll	5.00.2181.1	129.77 KB (132,880 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\dnsapi.dll			
wldap32.dll	5.00.2168.1	155.77 KB (159,504 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\wldap32.dll			
ws2help.dll	5.00.2134.1	17.77 KB (18,192 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\ws2help.dll			
ws2_32.dll	5.00.2134.1	69.77 KB (71,440 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\ws2_32.dll			
samlib.dll	5.00.2160.1	46.27 KB (47,376 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\samlib.dll			
netrap.dll	5.00.2134.1	11.27 KB (11,536 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\netrap.dll			
netapi32.dll	5.00.2194.1	302.77 KB (310,032 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\netapi32.dll			
profmap.dll	5.00.2181.1	29.27 KB (29,968 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\profmap.dll			
secur32.dll	5.00.2154.1	46.77 KB (47,888 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\secur32.dll			
sfc.dll	5.00.2164.1	84.27 KB (86,288 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\sfc.dll			
nddeapi.dll	5.00.2137.1	15.27 KB (15,632 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\nddeapi.dll			
userenv.dll	5.00.2185.1	361.27 KB (369,936 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\userenv.dll			
user32.dll	5.00.2180.1	393.27 KB (402,704 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\user32.dll			
gdi32.dll	5.00.2180.1	228.77 KB (234,256 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\gdi32.dll			
rpcrt4.dll	5.00.2193.1	434.27 KB (444,688 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\rpcrt4.dll			
advapi32.dll	5.00.2191.1	349.27 KB (357,648 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\advapi32.dll			
kernel32.dll	5.00.2191.1	715.27 KB (732,432 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\kernel32.dll			
msvcrt.dll	6.10.8637.0	288.09 KB (295,000 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\msvcrt.dll			
winlogon.exe	5.00.2182.1	173.27 KB (177,424 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\winlogon.exe			
sfcfiles.dll	5.00.2195.1	973.27 KB (996,624 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\sfcfiles.dll			
ntdll.dll	5.00.2163.1	469.77 KB (481,040 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\ntdll.dll			
smss.exe	5.00.2170.1	44.27 KB (45,328 bytes)	12/7/1999 4:00:00 AM	Microsoft Corporation
	c:\winnt\system32\smss.exe			

[Services]

Display Name	Name	State	Start Mode	Service Type	Path	Error Control	Start Name	Tag ID
Alerter	Alerter	Stopped	Manual	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
Application Management	AppMgmt	Stopped	Manual	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
Computer Browser	Browser	Stopped	Disabled	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
Indexing Service	cisvc	Stopped	Manual	Share Process	c:\winnt\system32\cisvc.exe			
	Normal	LocalSystem	0					
ClipBook	ClipSrv	Stopped	Manual	Own Process	c:\winnt\system32\clipsrv.exe		Normal	
	LocalSystem	0						
Distributed File System	Dfs	Stopped	Disabled	Own Process	c:\winnt\system32\dfssvc.exe			
	Normal	LocalSystem	0					
DHCP Client	Dhcp	Stopped	Disabled	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					

APPENDIX C – TUNABLE PARAMETERS

DHCP Server DHCPServer Stopped Disabled Share Process c:\winnt\system32\tcpsvcs.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 Server DNS Running Auto Own Process c:\winnt\system32\dns.exe Normal LocalSystem 0
DNS Client Dnscache Stopped Disabled 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
NetRAID-4M Remote Services Agent HPN_AGENT Running Auto Own Process c:\program
files\hp_raid\hpn\afaagent.exe Normal LocalSystem 0
Internet Authentication Service IAS Running Auto Share Process
c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
IIS Admin Service IISADMIN Running Auto Share Process
c:\winnt\system32\inetsrv\inetinfo.exe Normal LocalSystem 0
Intersite Messaging IsmServ Stopped Disabled Own Process c:\winnt\system32\ismserv.exe
Normal LocalSystem 0
Kerberos Key Distribution Center kdc Stopped Disabled Share Process
c:\winnt\system32\lsass.exe Normal LocalSystem 0
Server lanmanserver Running Auto Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0
Workstation lanmanworkstation Running Auto Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0
License Logging Service LicenseService Stopped Disabled 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 Stopped Disabled 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
FTP Publishing Service MSFTPSVC Stopped Disabled Share Process
c:\winnt\system32\inetsrv\inetinfo.exe Normal LocalSystem 0
Windows Installer MSIServer Stopped Manual Share Process c:\winnt\system32\msiexec.exe /v
Normal LocalSystem 0
Microsoft Search MSSEARCH Running Auto Share Process "c:\program files\common
files\system\mssearch\bin\mssearch.exe" Normal LocalSystem 0
MSSQLSERVER MSSQLSERVER Stopped Manual Own Process
c:\progra~1\micros~3\mssql\binn\sqlservr.exe Normal LocalSystem 0
MSSQLServerADHelper MSSQLServerADHelper Stopped Manual Own Process c:\program files\microsoft sql
server\80\tools\binn\sqladhlp.exe Normal LocalSystem 0
Network DDE NetDDE Stopped Manual Share Process c:\winnt\system32\netdde.exe
Normal LocalSystem 0
Network DDE DSDM NetDDEdsdm Stopped Manual Share Process
c:\winnt\system32\netdde.exe Normal LocalSystem 0
Net Logon Netlogon Stopped Manual Share Process c:\winnt\system32\lsass.exe
Normal LocalSystem 0
Network Connections Netman Running Manual Share Process c:\winnt\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
File Replication NtFrs Stopped Manual Own Process c:\winnt\system32\ntfrs.exe Ignore
LocalSystem 0
NT LM Security Support Provider NtLmSsp Running Manual Share Process
c:\winnt\system32\lsass.exe Normal LocalSystem 0
Removable Storage NtmsSvc Stopped Disabled 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 Stopped Disabled Share Process c:\winnt\system32\lsass.exe
Normal LocalSystem 0
Protected Storage ProtectedStorage Running Auto Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Remote Access Auto Connection Manager RasAuto Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Remote Access Connection Manager RasMan Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Routing and Remote Access RemoteAccess Stopped Disabled Share Process
c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Remote Registry Service RemoteRegistry Stopped Disabled Own Process c:\winnt\system32\regsvc.exe
Normal LocalSystem 0
Remote Procedure Call (RPC) Locator RpcLocator Running Manual Own Process c:\winnt\system32\locator.exe
Normal LocalSystem 0
Remote Procedure Call (RPC) RpcSs Running Auto Share Process
c:\winnt\system32\svchost -k rpcss Normal LocalSystem 0
QoS Admission Control (RSVP) RSVP Running Auto Own Process c:\winnt\system32\rsvp.exe -s
Normal LocalSystem 0
Security Accounts Manager SamSs Running Auto Share Process
c:\winnt\system32\lsass.exe Normal LocalSystem 0
Smart Card Helper SCardDrv Stopped Manual Share Process c:\winnt\system32\scardsvr.exe
Ignore LocalSystem 0
Smart Card SCardSvr Stopped Manual Share Process c:\winnt\system32\scardsvr.exe Ignore
LocalSystem 0
Task Scheduler Schedule Stopped Manual Share Process c:\winnt\system32\mstask.exe
Normal LocalSystem 0
RunAs Service seclogon Stopped Manual Share Process c:\winnt\system32\services.exe
Ignore LocalSystem 0
System Event Notification SENS Running Auto Share Process c:\winnt\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Internet Connection Sharing SharedAccess Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Simple TCP/IP Services SimpTcp Running Auto Share Process c:\winnt\system32\tcpsvcs.exe
Normal LocalSystem 0
Simple Mail Transport Protocol (SMTP) SMTPSVC Running Auto Share Process
c:\winnt\system32\inetsrv\inetinfo.exe 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 Stopped Disabled Own Process c:\winnt\system32\spoolsv.exe Normal
LocalSystem 0
SQLSERVERAGENT SQLSERVERAGENT Stopped Manual Own Process
c:\progra~1\micros~3\mssql\binn\sqlagent.exe Normal .\Administrator 0
Performance Logs and Alerts SysmonLog Stopped Manual Own Process c:\winnt\system32\smlogsvc.exe
Normal LocalSystem 0

APPENDIX C – TUNABLE PARAMETERS

Telephony TapiSrv Running Manual Share Process c:\winnt\system32\svchost.exe -k tapisrv
Normal LocalSystem 0
Terminal Services TermService Stopped Disabled Own Process c:\winnt\system32\termsrv.exe
Normal LocalSystem 0
Telnet TlntSvr Stopped Manual Own Process c:\winnt\system32\tlntsvr.exe Normal
LocalSystem 0
Distributed Link Tracking Server TrkSvr Stopped Manual Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Distributed Link Tracking Client TrkWks Stopped Disabled 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 Stopped Manual Share Process
c:\winnt\system32\inetsrv\inetinfo.exe Normal LocalSystem 0
Windows Management Instrumentation WinMgmt Running Auto Own Process
c:\winnt\system32\wbem\winmgmt.exe Ignore LocalSystem 0
Windows Internet Name Service (WINS) WINS Running Auto Own Process c:\winnt\system32\wins.exe
Normal LocalSystem 0
Windows Management Instrumentation Driver Extensions Wmi Running Manual Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0

[Program Groups]

Group Name Name User Name
Accessories Default User:Accessories Default User
Accessories\Accessibility Default User:Accessories\Accessibility Default User
Accessories\Entertainment Default User:Accessories\Entertainment Default User
Accessories\System Tools Default User:Accessories\System Tools Default User
Startup Default User:Startup Default User
Accessories All Users:Accessories All Users
Accessories\Accessibility All Users:Accessories\Accessibility All Users
Accessories\Communications All Users:Accessories\Communications All Users
Accessories\Entertainment All Users:Accessories\Entertainment All Users
Accessories\Games All Users:Accessories\Games All Users
Accessories\System Tools All Users:Accessories\System Tools All Users
Administrative Tools All Users:Administrative Tools All Users
HP_NetRAID-4M All Users:HP_NetRAID-4M All Users
Microsoft SQL Server All Users:Microsoft SQL Server All Users
Microsoft SQL Server - Switch All Users:Microsoft SQL Server - Switch All Users
Microsoft Visual Studio 6.0 All Users:Microsoft Visual Studio 6.0 All Users
Microsoft Visual Studio 6.0\Microsoft Visual Studio 6.0 Tools All Users:Microsoft Visual Studio 6.0\Microsoft Visual Studio 6.0 Tools All Users
MKS Toolkit All Users:MKS Toolkit All Users
Startup All Users:Startup All Users
VNC All Users:VNC All Users
VNC\Administrative Tools All Users:VNC\Administrative Tools All Users
Accessories PRF_SUT6\Administrator:Accessories PRF_SUT6\Administrator
Accessories\Accessibility PRF_SUT6\Administrator:Accessories\Accessibility PRF_SUT6\Administrator
Accessories\Entertainment PRF_SUT6\Administrator:Accessories\Entertainment PRF_SUT6\Administrator
Accessories\System Tools PRF_SUT6\Administrator:Accessories\System Tools PRF_SUT6\Administrator
Microsoft Web Publishing PRF_SUT6\Administrator:Microsoft Web Publishing PRF_SUT6\Administrator
Startup PRF_SUT6\Administrator:Startup PRF_SUT6\Administrator

[Startup Programs]

Program Command User Name Location
moderate moderate.cmd 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
Image Document "C:\Program Files\Windows NT\Accessories\ImageVue\KodakImg.exe"
WordPad Document "%ProgramFiles%\Windows NT\Accessories\WORDPAD.EXE"
Windows Media Services DRM Storage object Not Available
Bitmap Image C:\WINNT\System32\mspaint.exe

[Internet Explorer 5]

[Following are sub-categories of this main category]

[Summary]

Item Value
Version 5.00.2920.0000
Build 52920
Product ID 51879-270-6637794-05232
Application Path C:\Program Files\Internet Explorer
Language English (United States)
Active Printer Not Available

Cipher Strength 56-bit
Content Advisor Disabled
IEAK Install No

[File Versions]

File Version Size Date Path Company
advapi32.dll 5.0.2191.1 349 KB 12/7/1999 4:00:00 AM C:\WINNT\system32 Microsoft Corporation
advpack.dll 5.0.2920.0 87 KB 12/7/1999 4:00:00 AM C:\WINNT\system32 Microsoft Corporation
browselc.dll 5.0.2920.0 35 KB 12/7/1999 4:00:00 AM C:\WINNT\system32 Microsoft Corporation
browseui.dll 5.0.2920.0 793 KB 12/7/1999 4:00:00 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 5.81.2920.0 540 KB 12/7/1999 4:00:00 AM C:\WINNT\system32 Microsoft Corporation
crypt32.dll 5.131.2173.1 466 KB 12/7/1999 4:00:00 AM C:\WINNT\system32 Microsoft Corporation
enhsig.dll <File Missing> Not Available Not Available Not Available Not Available
iemigrat.dll <File Missing> Not Available Not Available Not Available Not Available
iesetup.dll 5.0.2920.0 57 KB 12/7/1999 4:00:00 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 5.0.2195.1 125 KB 12/7/1999 4:00:00 AM C:\WINNT\system32 Microsoft Corporation
imghelp.dll <File Missing> Not Available Not Available Not Available Not Available
inseng.dll 5.0.2920.0 72 KB 12/7/1999 4:00:00 AM C:\WINNT\system32 Microsoft Corporation

APPENDIX C – TUNABLE PARAMETERS

jobexec.dll	5.0.0.1	47 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
jscript.dll	5.1.0.4615	476 KB	12/7/1999 4:00:00 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 Not Available
mshtml.dll	5.0.2920.0	2302 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
msjava.dll	5.0.3234.0	918 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
msoss.dll	<File Missing>		Not Available	Not Available	Not Available Not Available
msxml.dll	5.0.2920.0	521 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
occache.dll	5.0.2920.0	86 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
ole32.dll	5.0.2181.1	966 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
oleaut32.dll	2.40.4512.1	600 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
olepro32.dll	5.0.4512.1	160 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
rsabase.dll	5.0.2150.1	129 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
rsaenh.dll	<File Missing>		Not Available	Not Available	Not Available Not Available
rsapi32.dll	<File Missing>		Not Available	Not Available	Not Available Not Available
rsasig.dll	<File Missing>		Not Available	Not Available	Not Available Not Available
schannel.dll	5.0.2170.0	140 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
shdoc401.dll	<File Missing>		Not Available	Not Available	Not Available Not Available
shdocvw.dll	5.0.2920.0	1078 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
shell32.dll	5.0.2920.0	2297 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
shlwapi.dll	5.0.2920.0	283 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
url.dll	5.0.2920.0	82 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
urlmon.dll	5.0.2920.0	427 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
vbscript.dll	5.1.0.4615	428 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
webcheck.dll	5.0.2920.0	252 KB	12/7/1999 4:00:00 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	5.0.2920.0	457 KB	12/7/1999 4:00:00 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	5.131.2143.1	162 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
wsock.vxd	<File Missing>		Not Available	Not Available	Not Available Not Available
wsock32.dll	5.0.2152.1	21 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
wsock32n.dll	<File Missing>		Not Available	Not Available	Not Available Not Available

[Connectivity]

Item	Value
Connection Preference	Never dial
EnableHttp1.1	1
ProxyHttp1.1	0

LAN Settings

AutoConfigProxy	wininet.dll
AutoProxyDetectMode	Disabled
AutoConfigURL	
Proxy	Enabled
ProxyServer	nsd-prx1.cup.hp.com:8088
ProxyOverride	

[Cache]

[Following are sub-categories of this main category]

[Summary]

Item	Value
Page Refresh Type	Automatic
Temporary Internet Files Folder	C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files
Total Disk Space	8667 MB
Available Disk Space	5533 MB
Maximum Cache Size	270 MB
Available Cache Size	269 MB

[List of Objects]

Program File Status	CodeBase
No cached object information available	

[Content]

[Following are sub-categories of this main category]

[Summary]

Item	Value
Content Advisor	Disabled

[Personal Certificates]

Issued To	Issued By	Validity	Signature Algorithm
Administrator	Administrator	10/30/2000 to 10/6/2100	sha1RSA

[Other People Certificates]

Issued To	Issued By	Validity	Signature Algorithm
No other people certificate information available			

[Publishers]

Name
No publisher information available

[Security]

Zone	Security Level
Local intranet	Medium-low
Trusted sites	Low
Internet	Medium
Restricted sites	High

C.3 Microsoft SQL Server 8.0 Startup Parameters

```
sqlservr -x -c -T3502 –g100
```

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.
-g100	Reserve 100 MB for non-buffer pool allocations.

C.4 Microsoft SQL Server 8.0 Stack Size

The default stack size for Microsoft SQL Server was changed using the EDITBIN utility, which ships with Microsoft Visual C++. The command used to change the stack size is:
editbin /Stack: 131072 sqlservr.exe

This command is fully documented as an article in the Microsoft Knowledge Base on the Microsoft Web Site at www.microsoft.com/support

C.5 BOOT.INI

Two switches were added to the *boot.ini* file
?? /3gb to cause Windows 2000 Datacenter Enterprise Edition to allow 3GB of user and 1GB of kernel virtual address space, rather than the usual 2GB space for each
?? /pae to cause Windows 2000 to support more than 4 GB of physical memory.

C.6 User Rights Assignment

The Group Policy Editor of Windows 2000 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.7 Microsoft SQL Server 8.0 Configuration Parameters

Name	minimum	maximum	config_value	run_value
affinity mask	0	2147483647	255	255
allow updates	0	1	1	1
awe enabled	0	1	1	1
c2 audit mode	0	1	0	0
cost threshold for parallelism	0	32767	5	5
cursor threshold	-1	2147483647	-1	-1
default full-text language	0	2147483647	1033	1033
default language	0	9999	0	0
fill factor (%)	0	100	0	0
index create memory (KB)	704	2147483647	0	0
lightweight pooling	0	1	1	1
locks	5000	2147483647	0	0
max degree of parallelism	0	32	1	1
max server memory (MB)	4	2147483647	0	2147483647
max text repl size (B)	0	2147483647	65536	65536
max worker threads	32	32767	235	235
media retention	0	365	0	0
min memory per query (KB)	512	2147483647	1024	1024
min server memory (MB)	0	2147483647	100	100
nested triggers	0	1	1	1
network packet size (B)	512	65535	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	24	24
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	0	0
scan for startup procs	0	1	0	0
set working set size	0	1	0	0
show advanced options	0	1	1	1
two digit year cutoff	1753	9999	2049	2049
user connections	0	32767	0	0
user options	0	32767	0	0

C.8 Internal DAC Configuration Parameters

Two sets of registry entries were used with the performance driver.

1) DpcProcessors
   It specifies how many processors are available for Dpc processing.
2) AdapterAffinityMasks
   It assigns the handling of adapter interrupt to a particular processor. In a multiprocessor/multiadapter system with 8 processors and 7 adapters, one processor could service two or more adapter interrupts to get the balancing work load over the 8 processors. The adapter to processor affinitizing is specified by the adapter affinity mask registry entries.

The disk controllers used during this benchmark were configured as follows:

Key Name: SYSTEM\CurrentControlSet\Services\hpnsa
Class Name: <NO CLASS>
Last Write Time: 12/21/2000 - 2:51 PM
Value 0
Name: Adapt
Type: REG_DWORD
Data: 0x12

Value 1
Name: AdaptAffinityMask0
Type: REG_DWORD
Data: 0x12

Value 2
Name: AdaptAffinityMask1
Type: REG_DWORD
Data: 0x12

Value 3
Name: AdaptAffinityMask2
Type: REG_DWORD
Data: 0x24

Value 4
Name: AdaptAffinityMask3
Type: REG_DWORD
Data: 0x24

Value 5
Name: AdaptAffinityMask4
Type: REG_DWORD
Data: 0x81

Value 6
Name: AdaptAffinityMask5
Type: REG_DWORD
Data: 0x48

Value 7
Name: AdaptAffinityMask6
Type: REG_DWORD
Data: 0x81

Value 8
Name: DependOnGroup
Type: REG_MULTI_SZ
Data: SCSI miniport

Value 9
Name: DependOnService
Type: REG_MULTI_SZ
Data:

Value 10
Name: DisplayName
Type: REG_SZ
Data: Hewlett Packard NetRAID-4M Driver

Value 11
Name: DpcProcessors
Type: REG_DWORD
Data: 0x8

Value 12
Name: ErrorControl
Type: REG_DWORD
Data: 0x1

Value 13
Name: Group
Type: REG_SZ
Data: port

Value 14
Name: ImagePath
Type: REG_EXPAND_SZ
Data: System32\DRIVERS\hpnsa.sys

Value 15
Name: Start
Type: REG_DWORD
Data: 0

Value 16
Name: Tag
Type: REG_DWORD
Data: 0x1

Value 17
Name: Type
Type: REG_DWORD
Data: 0x1

Key Name: SYSTEM\CurrentControlSet\Services\hpnsa\Enum
Class Name: <NO CLASS>

APPENDIX C – TUNABLE PARAMETERS

Last Write Time: 12/21/2000 - 2:35 PM
Value 0
Name: 0
Type: REG_SZ
Data: PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&267a616a&0&20

Value 1
Name: 1
Type: REG_SZ
Data: PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&13c0b0c5&0&28

Value 2
Name: 2
Type: REG_SZ
Data: PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&13c0b0c5&0&30

Value 3
Name: 3
Type: REG_SZ
Data: PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&1070020&0&20

Value 4
Name: 4
Type: REG_SZ
Data: PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&1070020&0&28

Value 5
Name: 5
Type: REG_SZ
Data: PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&29e81982&0&20

Value 6
Name: 6
Type: REG_SZ
Data: PCI\VEN_1011&DEV_0046&SUBSYS_10C2103C&REV_01\3&29e81982&0&28

Value 7
Name: Count
Type: REG_DWORD
Data: 0x7

Value 8
Name: NextInstance
Type: REG_DWORD
Data: 0x7

Key Name: SYSTEM\CurrentControlSet\Services\hpnsa\Parameters
Class Name: <NO CLASS>
Last Write Time: 10/30/2000 - 9:33 AM

Key Name: SYSTEM\CurrentControlSet\Services\hpnsa\Parameters\PnpInterface
Class Name: <NO CLASS>
Last Write Time: 10/30/2000 - 9:33 AM
Value 0
Name: 5
Type: REG_DWORD
Data: 0x1

Key Name: SYSTEM\CurrentControlSet\Services\hpnsa\Security
Class Name: <NO CLASS>
Last Write Time: 10/30/2000 - 9:33 AM
Value 0
Name: Security
Type: REG_BINARY
Data:

```
00000000  01 00 14 80 a0 00 00 00 - ac 00 00 00 14 00 00 00  .... ...¬.......
00000010  30 00 00 00 02 00 1c 00 - 01 00 00 00 02 80 14 00  0...............
00000020  ff 01 0f 00 01 01 00 00 - 00 00 00 01 00 00 00 00  ÿ...............
00000030  02 00 70 00 04 00 00 00 - 00 00 18 00 fd 01 02 00  ..p.........ý...
00000040  01 01 00 00 00 00 00 05 - 12 00 00 00 50 00 5f 00  ............P._.
00000050  00 00 1c 00 ff 01 0f 00 - 01 02 00 00 00 00 00 05  ....ÿ...........
00000060  20 00 00 00 20 02 00 00 - 32 00 66 00 00 00 18 00   ... ...2.f.....
00000070  8d 01 02 00 01 01 00 00 - 00 00 00 05 0b 00 00 00  ................
00000080  20 02 00 00 00 00 1c 00 - fd 01 02 00 01 02 00 00   .......ý.......
00000090  00 00 00 05 20 00 00 00 - 23 02 00 00 32 00 66 00  .... ...#...2.f.
000000a0  01 01 00 00 00 00 00 05 - 12 00 00 00 01 01 00 00  ................
000000b0  00 00 00 05 12 00 00 00 -                          ........
```

C.9 Client System Configuration Parameters

COM+ Settings

TPCC.AllTxns:
Activation:
Enable Object Pooling selected
Minimum Pool Size: 30
Maximum Pool Size: 30
Creation Timeout: 60000
Enable Object Construction
Enable Just In Time Activation
Concurrency:
Concurrency Required

Microsoft IIS Registry Parameters

Key Name: SYSTEM\CurrentControlSet\Services\InetInfo\Parameters
Class Name: <NO CLASS>
Last Write Time: 6/26/2000 - 3:40 PM
Value 0
Name: DispatchEntries
Type: REG_MULTI_SZ
Data: LDAPSVC

Value 1
Name: ListenBackLog
Type: REG_DWORD
Data: 0x30

Value 2
Name: PoolThreadLimit
Type: REG_DWORD
Data: 0x400
Value 3
Name: ThreadTimeout
Type: REG_DWORD
Data: 0x15180

World Wide Web Service Registry Parameters

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters
Class Name: <NO CLASS>
Last Write Time: 11/9/1999 - 5:44 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: 11/9/1999 - 2:53 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory
Class Name: <NO CLASS>
Last Write Time: 11/9/1999 - 2:53 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory
Class Name: <NO CLASS>
Last Write Time: 11/9/1999 - 2:53 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map
Class Name: <NO CLASS>
Last Write Time: 11/9/1999 - 3:05 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots
Class Name: <NO CLASS>
Last Write Time: 9/26/2000 - 3:30 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,,205

Value 2
Name: /IISAdmin
Type: REG_SZ
Data: C:\WINNT\System32\inetsrv\iisadmin,,201

Value 3
Name: /IISHelp
Type: REG_SZ
Data: c:\winnt\help\iishelp,,201

Value 4
Name: /IISSamples
Type: REG_SZ
Data: c:\inetpub\iissamples,,201

Value 5
Name: /MSADC
Type: REG_SZ
Data: c:\program files\common files\system\msadc,,205

Value 6
Name: /Printers
Type: REG_SZ
Data: C:\WINNT\web\printers,,201

Value 7
Name: /Rpc
Type: REG_SZ
Data: C:\WINNT\System32\RpcProxy,,4

Value 8
Name: /Scripts
Type: REG_SZ
Data: c:\inetpub\scripts,,204

TPCC Application Registry Parameters

Key Name: SOFTWARE\Microsoft\TPCC
Class Name: <NO CLASS>
Last Write Time: 6/23/2000 - 1:13 PM
Value 0
Name: COM_SinglePool
Type: REG_SZ
Data: YES

Value 1
Name: DB_Protocol
Type: REG_SZ
Data: DBLIB

Value 2
Name: DbName
Type: REG_SZ
Data: tpcc

Value 3
Name: DbPassword
Type: REG_SZ
Data:

Value 4
Name: DbServer
Type: REG_SZ
Data: prf_sut6

Value 5
Name: DbUser
Type: REG_SZ
Data: sa

Value 6
Name: MaxConnections
Type: REG_DWORD
Data: 0x1f40

Value 7
Name: MaxPendingDeliveries
Type: REG_DWORD
Data: 0x3e8

Value 8
Name: NumberOfDeliveryThreads
Type: REG_DWORD
Data: 0x5

Value 9
Name: Path
Type: REG_SZ
Data: c:\inetpub\wwwroot\

Value 10
Name: TxnMonitor
Type: REG_SZ
Data: COM

Microsoft Windows 2000 Server Configuration Parameters

System Information report written at: 09/05/2000 12:57:04 PM
[System Information]

[Following are sub-categories of this main category]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Server
Version	5.0.2128 Build 2128
OS Manufacturer	Microsoft Corporation
System Name	N1_CLIENT1
System Manufacturer	Hewlett Packard
System Model	HP NetServer
System Type	X86-based PC
Processor	x86 Family 6 Model 7 Stepping 3 GenuineIntel ~600 Mhz
BIOS Version	09/16/99
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition1
Locale	United States
User Name	N1_CLIENT1\Administrator
Time Zone	Pacific Daylight Time
Total Physical Memory	523,756 KB
Available Physical Memory	385,488 KB
Total Virtual Memory	1,802,756 KB
Available Virtual Memory	1,555,960 KB
Page File Space	1,279,000 KB
Page File	Not Available

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

Resource	Device
IRQ 19	Intel 82371AB/EB PCI to USB Universal Host Controller
IRQ 19	HP NetServer 10/100TX PCI LAN Adapter #2
IRQ 18	Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller
IRQ 18	Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller

[DMA]

Channel	Device	Status
4	Direct memory access controller	OK
2	Standard floppy disk controller	OK

APPENDIX C – TUNABLE PARAMETERS

[Forced Hardware]

Device PNP Device ID
No Forced Hardware

[I/O]

Address Range	Device	Status
0x0000-0x0CF7	PCI bus	OK
0x0000-0x0CF7	Direct memory access controller	OK
0x0D00-0xFFFF	PCI bus	OK
0x9000-0x9FFF	Intel 82443BX Pentium(r) II Processor to AGP Controller	OK
0x9000-0x9FFF	ATI Technologies Inc. 3D RAGE IIC AGP	OK
0x03B0-0x03BB	Intel 82443BX Pentium(r) II Processor to AGP Controller	OK
0x03B0-0x03BB	ATI Technologies Inc. 3D RAGE IIC AGP	OK
0x03C0-0x03DF	Intel 82443BX Pentium(r) II Processor to AGP Controller	OK
0x03C0-0x03DF	ATI Technologies Inc. 3D RAGE IIC AGP	OK
0x0A79-0x0A79	ISAPNP Read Data Port	OK
0x0279-0x0279	ISAPNP Read Data Port	OK
0x0274-0x0277	ISAPNP Read Data Port	OK
0x0081-0x008F	Direct memory access controller	OK
0x00C0-0x00DF	Direct memory access controller	OK
0x0070-0x0071	System CMOS/real time clock	OK
0x0020-0x0021	Programmable interrupt controller	OK
0x00A0-0x00A1	Programmable interrupt controller	OK
0x00F0-0x00FE	Numeric data processor	OK
0x0040-0x0043	System timer	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
0x03F0-0x03F5	Standard floppy disk controller	OK
0x03F7-0x03F7	Standard floppy disk controller	OK
0x03F8-0x03FF	Communications Port (COM1)	OK
0x0378-0x037B	Printer Port (LPT1)	OK
0x02F8-0x02FF	Communications Port (COM2)	OK
0x1880-0x188F	Intel(r) 82371AB/EB PCI Bus Master 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
0x1840-0x185F	Intel 82371AB/EB PCI to USB Universal Host Controller	OK
0x1000-0x10FF	Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller	OK
0x1400-0x14FF	Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller	OK
0x1800-0x183F	HP NetServer 10/100TX PCI LAN Adapter #2	OK
0x1860-0x187F	HP NetServer 10/100TX PCI LAN Adapter	OK

[IRQs]

IRQ Number	Device
20	Microsoft ACPI-Compliant System
8	System CMOS/real time clock
13	Numeric data processor
0	System timer
1	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
12	PS/2 Compatible Mouse
6	Standard floppy disk controller
4	Communications Port (COM1)
3	Communications Port (COM2)
14	Primary IDE Channel
15	Secondary IDE Channel
19	Intel 82371AB/EB PCI to USB Universal Host Controller
19	HP NetServer 10/100TX PCI LAN Adapter #2
18	Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller
18	Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller
16	HP NetServer 10/100TX PCI LAN Adapter

[Memory]

Range	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	Intel 82443BX Pentium(r) II Processor to AGP Controller	OK
0xA0000-0xBFFFF	ATI Technologies Inc. 3D RAGE IIC AGP	OK
0xC0000-0xC3FFF	PCI bus	OK
0xC4000-0xC7FFF	PCI bus	OK
0xC8000-0xCBFFF	PCI bus	OK
0xCC000-0xCFFFF	PCI bus	OK
0xE8000-0xEBFFF	PCI bus	OK
0xEC000-0xEFFFF	PCI bus	OK
0x20000000-0xFFDFFFFF	PCI bus	OK
0xFC300000-0xFC3FFFFF	Intel 82443BX Pentium(r) II Processor to AGP Controller	OK
0xFC300000-0xFC3FFFFF	ATI Technologies Inc. 3D RAGE IIC AGP	OK
0xFD000000-0xFDFFFFFF	Intel 82443BX Pentium(r) II Processor to AGP Controller	OK
0xFD000000-0xFDFFFFFF	ATI Technologies Inc. 3D RAGE IIC AGP	OK
0xE0000000-0xEFFFFFFF	Intel 82443BX Pentium(r) II Processor to AGP Controller	OK
0xFFFD0000-0xFFFEFFFF	System board	OK
0xFC200000-0xFC200FFF	Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller	OK
0xFC201000-0xFC201FFF	Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller	OK
0xFC202000-0xFC202FFF	HP NetServer 10/100TX PCI LAN Adapter #2	OK
0xFC000000-0xFC0FFFFF	HP NetServer 10/100TX PCI LAN Adapter #2	OK
0xFC203000-0xFC203FFF	HP NetServer 10/100TX PCI LAN Adapter	OK
0xFC100000-0xFC1FFFFF	HP NetServer 10/100TX PCI LAN 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)	9/9/1999 5:00:00 PM			
c:\winnt\system32\msg723.acm	Microsoft Corporation		OK				
	C:\WINNT\System32\MSG723.ACM	4.4.3385	106.77 KB (109,328 bytes)	11/9/1999 1:53:30 PM			
c:\winnt\system32\lhacm.acm	Microsoft Corporation		OK				
	C:\WINNT\System32\LHACM.ACM	4.4.3385	33.27 KB (34,064 bytes)	11/9/1999 1:53:30 PM			
c:\winnt\system32\tssoft32.acm	DSP GROUP, INC.		OK				
	C:\WINNT\System32\TSSOFT32.ACM	1.01	9.27 KB (9,488 bytes)	9/9/1999 5:00:00 PM			
c:\winnt\system32\msgsm32.acm	Microsoft Corporation		OK				
	C:\WINNT\System32\MSGSM32.ACM	5.00.2113.1	22.27 KB (22,800 bytes)	9/9/1999 5:00:00 PM			

APPENDIX C – TUNABLE PARAMETERS

c:\winnt\system32\msg711.acm Microsoft Corporation OK
C:\WINNT\System32\MSG711.ACM 5.00.2113.1 10.27 KB (10,512 bytes) 9/9/1999 5:00:00 PM
c:\winnt\system32\msadp32.acm Microsoft Corporation OK
C:\WINNT\System32\MSADP32.ACM 5.00.2113.1 14.77 KB (15,120 bytes) 9/9/1999 5:00:00 PM
c:\winnt\system32\imaadp32.acm Microsoft Corporation OK
C:\WINNT\System32\IMAADP32.ACM 5.00.2113.1 16.27 KB (16,656 bytes) 9/9/1999 5:00:00 PM

[Video Codecs]

Codec Manufacturer Description Status File Version Size Creation Date
c:\winnt\system32\msh261.drv Microsoft Corporation OK
C:\WINNT\System32\MSH261.DRV 4.4.3385 163.77 KB (167,696 bytes) 11/9/1999 1:53:30 PM
c:\winnt\system32\msh263.drv Microsoft Corporation OK
C:\WINNT\System32\MSH263.DRV 4.4.3385 252.27 KB (258,320 bytes) 11/9/1999 1:52:59 PM
c:\winnt\system32\msvidc32.dll Microsoft Corporation OK
C:\WINNT\System32\MSVIDC32.DLL 5.00.2113.1 27.27 KB (27,920 bytes) 9/9/1999 5:00:00 PM
c:\winnt\system32\msrle32.dll Microsoft Corporation OK
C:\WINNT\System32\MSRLE32.DLL 5.00.2113.1 10.77 KB (11,024 bytes) 9/9/1999 5:00:00 PM
c:\winnt\system32\ir32_32.dll Intel(R) Corporation OK
C:\WINNT\System32\IR32_32.DLL Not Available 194.50 KB (199,168 bytes) 9/9/1999 5:00:00 PM
c:\winnt\system32\iccvid.dll Radius Inc. OK C:\WINNT\System32\ICCVID.DLL
1.10.0.6 108.00 KB (110,592 bytes) 9/9/1999 5:00:00 PM
c:\winnt\system32\ir50_32.dll Intel Corporation Indeo® video 5.10 OK
C:\WINNT\System32\IR50_32.DLL R.5.10.15.2.54 737.50 KB (755,200 bytes) 9/9/1999 5:00:00 PM

[CD-ROM]

Item Value
Drive D:
Description CD-ROM Drive
Media Loaded False
Media Type CD-ROM
Name HITACHI CDR-8435
Manufacturer (Standard CD-ROM drives)
Status Unknown
Transfer Rate Not Available
SCSI Target ID 0
PNP Device ID IDE\CDROMHITACHI_CDR-8435_____0010____\5&32611E33&0&0.0.0

[Sound Device]

Item Value
No sound devices

[Display]

Item Value
Name ATI Technologies Inc. 3D RAGE IIC AGP
PNP Device ID PCI\VEN_1002&DEV_475A&SUBSYS_00000000&REV_7A\4&415A68E&0&0008
Adapter Type ATI 3D RAGE IIC AGP (A21), ATI Technologies Inc. compatible
Adapter Description ATI Technologies Inc. 3D RAGE IIC AGP
Adapter RAM 4.00 MB (4,194,304 bytes)
Installed Drivers atirage.sys
Driver Version 5.00.2112.1
INF File display.inf (atirage section)
Color Planes 1
Color Table Entries 65536
Resolution 1024 x 768 x 75 hertz
Bits/Pixel 16

[Infrared]

Item Value
No infrared devices

[Input]

[Following are sub-categories of this main category]

[Keyboard]

Item Value
Description Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
Name Enhanced (101- or 102-key)
Layout 00000409
PNP Device ID ACPI\PNP0303\4&244B3C61&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&244B3C61&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 [000] HP NetServer 10/100TX PCI LAN Adapter
Adapter Type Ethernet 802.3
Product Name HP NetServer 10/100TX PCI LAN Adapter
Installed True
PNP Device ID PCI\VEN_8086&DEV_1229&SUBSYS_10C3103C&REV_05\3&61AAA01&0&50
Last Reset 9/5/2000 5:26:16 AM
Index 0
Service Name HPTX

APPENDIX C – TUNABLE PARAMETERS

IP Address 180.20.1.100
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:90:27:9C:C7:E8
Service Name HPTX
IRQ Number 16
I/O Port 0x1860-0x187F
Driver 83.27 KB (85,264 bytes)

Name [001] HP NetServer 10/100TX PCI LAN Adapter
Adapter Type Ethernet 802.3
Product Name HP NetServer 10/100TX PCI LAN Adapter
Installed True
PNP Device ID PCI\VEN_8086&DEV_1229&SUBSYS_10CA103C&REV_08\3&61AAA01&0&30
Last Reset 9/5/2000 5:26:16 AM
Index 1
Service Name HPTX
IP Address 15.75.206.171
IP Subnet 255.255.248.0
Default IP Gateway 15.75.200.1
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:E0:18:C1:CF:FF
Service Name HPTX
IRQ Number 19
I/O Port 0x1800-0x183F
Driver 83.27 KB (85,264 bytes)

Name [002] RAS Async Adapter
Adapter Type RAS Async Adapter
Product Name RAS Async Adapter
Installed True
PNP Device ID Not Available
Last Reset 9/5/2000 5:26:16 AM
Index 2
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Not Available

Name [003] WAN Miniport (L2TP)
Adapter Type WAN Miniport (L2TP)
Product Name WAN Miniport (L2TP)
Installed True
PNP Device ID ROOT\MS_L2TPMINIPORT\0000
Last Reset 9/5/2000 5:26:16 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 48.11 KB (49,264 bytes)

Name [004] 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 9/5/2000 5:26:16 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 Not Available
Service Name PptpMiniport
Driver 44.58 KB (45,648 bytes)

Name [005] Direct Parallel
Adapter Type Direct Parallel
Product Name Direct Parallel
Installed True
PNP Device ID ROOT\MS_PTIMINIPORT\0000
Last Reset 9/5/2000 5:26:16 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 16.48 KB (16,880 bytes)

Name [006] WAN Miniport (IP)
Adapter Type WAN Miniport (IP)
Product Name WAN Miniport (IP)
Installed True
PNP Device ID ROOT\MS_NDISWANIP\0000

APPENDIX C – TUNABLE PARAMETERS

Last Reset 9/5/2000 5:26:16 AM
Index 6
Service Name NdisWan
IP Address 0.0.0.0
IP Subnet 0.0.0.0
Default IP Gateway
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 86.92 KB (89,008 bytes)

[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
SupportsFragmentation Not Available
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
SupportsFragmentation Not Available
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
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting True

Name RSVP TCP Service Provider
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 16 bytes
MaximumMessageSize 0 bytes
MessageOriented False
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption True
SupportsExpeditedData True
SupportsFragmentation Not Available
SupportsGracefulClosing True
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{D428C674-0F6C-4AF5-A20B-B66B08421706}] 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
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{D428C674-0F6C-4AF5-A20B-B66B08421706}] DATAGRAM 0

APPENDIX C – TUNABLE PARAMETERS

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
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{43466D29-0F72-45B2-AFA4-2ADFA67EDCF1}] 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
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{43466D29-0F72-45B2-AFA4-2ADFA67EDCF1}] 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
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{FD8C9E7C-5A63-4B52-8837-748AD106BD8A}] 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
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{FD8C9E7C-5A63-4B52-8837-748AD106BD8A}] DATAGRAM 2
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{2FE462C9-7B2D-4C11-9639-D5362D3BD3A4}] 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
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{2FE462C9-7B2D-4C11-9639-D5362D3BD3A4}] DATAGRAM 3

APPENDIX C – TUNABLE PARAMETERS

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
SupportsFragmentation	Not Available
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	5.00.2120.1
Size	21.27 KB (21,776 bytes)

[Ports]

[Following are sub-categories of this main category]

[Serial]

Item	Value
Name	Communications Port (COM1)
Status	OK
PNP Device ID	ACPI\PNP0501\1
Maximum Input Buffer Size	0
Maximum Output Buffer Size	False
Settable Baud Rate	True
Settable Data Bits	True
Settable Flow Control	True
Settable Parity	True
Settable Parity Check	True
Settable Stop Bits	True
Settable RLSD	True
Supports RLSD	True
Supports 16 Bit Mode	False
Supports Special Characters	False
Baud Rate	9600
Bits/Byte	8
Stop Bits	1
Parity	NONE
Busy	0
Abort Read/Write on Error	0
Binary Mode Enabled	-1
Continue XMit on XOff	0
CTS Outflow Control	0
Discard NULL Bytes	0
DSR Outflow Control	0
DSR Sensitivity	0
DTR Flow Control Type	ENABLE
EOF Character	0
Error Replace Character	0
Error Replacement Enabled	0
Event Character	0
Parity Check Enabled	0
RTS Flow Control Type	ENABLE
XOff Character	19
XOffXMit Threshold	512
XOn Character	17
XOnXMit Threshold	2048
XOnXOff InFlow Control	0
XOnXOff OutFlow Control	0
IRQ Number	4
I/O Port	0x03F8-0x03FF
Driver	60.95 KB (62,416 bytes)

Name	Communications Port (COM2)
Status	OK
PNP Device ID	ACPI\PNP0501\2
Maximum Input Buffer Size	0
Maximum Output Buffer Size	False
Settable Baud Rate	True
Settable Data Bits	True
Settable Flow Control	True
Settable Parity	True
Settable Parity Check	True
Settable Stop Bits	True
Settable RLSD	True
Supports RLSD	True
Supports 16 Bit Mode	False
Supports Special Characters	False
Baud Rate	9600
Bits/Byte	8
Stop Bits	1
Parity	NONE
Busy	0
Abort Read/Write on Error	0
Binary Mode Enabled	-1
Continue XMit on XOff	0
CTS Outflow Control	0
Discard NULL Bytes	0
DSR Outflow Control	0
DSR Sensitivity	0
DTR Flow Control Type	ENABLE
EOF Character	0
Error Replace Character	0
Error Replacement Enabled	0
Event Character	0

APPENDIX C – TUNABLE PARAMETERS

Parity Check Enabled 0
RTS Flow Control Type ENABLE
XOff Character 19
XOffXMit Threshold 512
XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control 0
IRQ Number 3
I/O Port 0x02F8-0x02FF
Driver 60.95 KB (62,416 bytes)

[Parallel]

Item Value
Name LPT1
PNP Device ID ACPI\PNP0400\1

[Storage]

[Following are sub-categories of this main category]

[Drives]

Item Value
Drive A:
Description 3 1/2 Inch Floppy Drive

Drive C:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 8.46 GB (9,088,901,120 bytes)
Free Space 3.59 GB (3,859,812,352 bytes)
Volume Name
Volume Serial Number 2C767FD4
Partition Disk #0, Partition #0
Partition Size 8.46 GB (9,088,902,144 bytes)
Starting Offset 32256 bytes
Drive Description Disk drive
Drive Manufacturer (Standard disk drives)
Drive Model HP 9.10GB A 68-SA40 SCSI Disk Device
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 1
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 2
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 9097159680 bytes
Drive TotalCylinders 1106
Drive TotalSectors 17767890
Drive TotalTracks 282030
Drive TracksPerCylinder 255

Drive L:
Description Network Connection
Provider Name \\nrddata\g$

[SCSI]

Item Value
Name Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller
Caption Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller
Driver aic78xx
Status OK
PNP Device ID PCI\VEN_9004&DEV_7895&SUBSYS_78959004&REV_04\3&61AAA01&0&28
Device ID PCI\VEN_9004&DEV_7895&SUBSYS_78959004&REV_04\3&61AAA01&0&28
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 18
I/O Port 0x1000-0x10FF
Driver 55.58 KB (56,912 bytes)

Name Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller
Caption Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller
Driver aic78xx
Status OK
PNP Device ID PCI\VEN_9004&DEV_7895&SUBSYS_78959004&REV_04\3&61AAA01&0&29
Device ID PCI\VEN_9004&DEV_7895&SUBSYS_78959004&REV_04\3&61AAA01&0&29
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 18
I/O Port 0x1400-0x14FF
Driver 55.58 KB (56,912 bytes)

[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
USB Root Hub USB\ROOT_HUB\4&5741930&0

[Software Environment]

[Following are sub-categories of this main category]

[Drivers]

APPENDIX C – TUNABLE PARAMETERS

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 Start		
	Running	OK	Normal	False	True				
acpiec	ACPIEC	c:\winnt\system32\drivers\acpiec.sys		Kernel Driver	False	Disabled	Stopped	OK	
	Normal	False	False						
adpu160m	adpu160m	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
afd	AFD Networking Support Environment			c:\winnt\system32\drivers\afd.sys		Kernel Driver	True	Auto	
Start	Running	OK	Normal	False	True				
agp440	AGP Bus Filter		c:\winnt\system32\drivers\agp440.sys		Kernel Driver	True	Boot Start		
	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	c:\winnt\system32\drivers\aic78xx.sys		Kernel Driver	True	Boot Start	Running	OK	
	Normal	False	True						
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								
arp1394	1394 ARP Client Protocol		c:\winnt\system32\drivers\arp1394.sys		Kernel Driver	False	Demand Start		
	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		
	Demand Start		Stopped	OK	Normal	False	False		
atapi	Standard IDE/ESDI Hard Disk Controller			c:\winnt\system32\drivers\atapi.sys		Kernel Driver	True	Boot	
Start	Running	OK	Normal	False	True				
atdisk	Atdisk	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Ignore	False
	False								
atirage	atirage	c:\winnt\system32\drivers\atirage.sys		Kernel Driver	True	Demand Start			
	Running	OK	Ignore	False	True				
atmarpc	ATM ARP Client Protocol		c:\winnt\system32\drivers\atmarpc.sys		Kernel Driver	False	Demand Start		
	Stopped	OK	Normal	False	False				
audstub	Audio Stub Driver		c:\winnt\system32\drivers\audstub.sys		Kernel Driver	True	Demand Start		
	Running	OK	Normal	False	True				
beep	Beep	c:\winnt\system32\drivers\beep.sys		Kernel Driver	True	System Start	Running	OK	
	Normal	False	True						
buslogic	BusLogic	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
cd20xmt	cd20xmt	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
cdaudio	Cdaudio	c:\winnt\system32\drivers\cdaudio.sys		Kernel Driver	False	System Start	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 Start		
	Running	OK	Normal	False	True				
changer	Changer	Not Available	Kernel Driver	False	System Start	Stopped	OK	Ignore	False
	False								
cpqarray	Cpqarray	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
cpqfcalm	cpqfcalm	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
cpqfws2e	cpqfws2e	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
dac960nt	dac960nt	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
deckzpsx	deckzpsx	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
dfsdriver	DfsDriver	c:\winnt\system32\drivers\dfs.sys		File System Driver		True	Boot Start		
	Running	OK	Normal	False	True				
disk	Disk Driver	c:\winnt\system32\drivers\disk.sys		Kernel Driver	True	Boot Start	Running	OK	
	Normal	False	True						
diskperf	Diskperf	c:\winnt\system32\drivers\diskperf.sys		Kernel Driver	True	Boot Start	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	
Start	Running	OK	Normal	False	True				
dmload	dmload	c:\winnt\system32\drivers\dmload.sys		Kernel Driver	True	Boot Start	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		
	Demand Start		Running	OK	Normal	False	True		
fireport	fireport	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
flashpnt	flashpnt	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
flpydisk	Floppy Disk Driver		c:\winnt\system32\drivers\flpydisk.sys		Kernel Driver	True	Demand Start		
	Running	OK	Normal	False	True				
ftdisk	Volume Manager Driver		c:\winnt\system32\drivers\ftdisk.sys		Kernel Driver	True	Boot Start		
	Running	OK	Normal	False	True				
gpc	Generic Packet Classifier		c:\winnt\system32\drivers\msgpc.sys		Kernel Driver	True	Demand Start		
	Running	OK	Normal	False	True				
hptx	HP NetServer 10/100TX PCI LAN Adapter Driver				c:\winnt\system32\drivers\hptxnt5.sys		Kernel Driver	True	
	Demand Start		Running	OK	Normal	False	True		
i8042prt	i8042 Keyboard and PS/2 Mouse Port Driver				c:\winnt\system32\drivers\i8042prt.sys		Kernel Driver	True	
	System Start	Running	OK	Normal	False	True			
ini910u	ini910u	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
intelide	Intellde	c:\winnt\system32\drivers\intelide.sys		Kernel Driver	True	Boot Start	Running	OK	
	Normal	False	True						
ipfilterdriver	IP Traffic Filter Driver		c:\winnt\system32\drivers\ipfltdrv.sys		Kernel Driver	False	Demand Start		
	Stopped	OK	Normal	False	False				

APPENDIX C – TUNABLE PARAMETERS

ipinip	IP in IP Tunnel Driver	c:\winnt\system32\drivers\ipinip.sys	Kernel Driver	False	Demand Start				
	Stopped	OK	Normal	False	False				
ipnat	IP Network Address Translator	c:\winnt\system32\drivers\ipnat.sys	Kernel Driver	False					
	Demand Start	Stopped	OK	Normal	False	False			
ipsec	IPSEC driver	c:\winnt\system32\drivers\ipsec.sys	Kernel Driver	True	Demand Start				
	Running	OK	Normal	False	True				
ipsraidn	ipsraidn	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
isapnp	PnP ISA/EISA Bus Driver	c:\winnt\system32\drivers\isapnp.sys	Kernel Driver	True	Boot Start				
	Running	OK	Critical	False	True				
kbdclass	Keyboard Class Driver	c:\winnt\system32\drivers\kbdclass.sys	Kernel Driver	True	System Start				
	Running	OK	Normal	False	True				
ksecdd	KSecDD	c:\winnt\system32\drivers\ksecdd.sys	Kernel Driver	True	Boot Start	Running	OK		
	Normal	False	True						
lbrtfdc	lbrtfdc	Not Available	Kernel Driver	False	System Start	Stopped	OK	Ignore	False
	False								
lp6nds35	lp6nds35	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
mnmdd	mnmdd	c:\winnt\system32\drivers\mnmdd.sys	Kernel Driver	True	System Start	Running	OK		
	Ignore	False	True						
modem	Modem	c:\winnt\system32\drivers\modem.sys	Kernel Driver	False	Demand Start				
	Stopped	OK	Ignore	False	False				
mouclass	Mouse Class Driver	c:\winnt\system32\drivers\mouclass.sys	Kernel Driver	True	System Start				
	Running	OK	Normal	False	True				
mountmgr	MountMgr	c:\winnt\system32\drivers\mountmgr.sys	Kernel Driver	True	Boot Start	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 Start			
	Running	OK	Normal	False	True				
msfs	Msfs	c:\winnt\system32\drivers\msfs.sys	File System Driver		True	System Start			
	Running	OK	Normal	False	True				
msskssrv	Microsoft Streaming Service Proxy	c:\winnt\system32\drivers\msskssrv.sys	Kernel Driver	False					
	Demand Start	Stopped	OK	Normal	False	False			
mspclock	Microsoft Streaming Clock Proxy	c:\winnt\system32\drivers\mspclock.sys	Kernel Driver	False					
	Demand Start	Stopped	OK	Normal	False	False			
mspqm	Microsoft Streaming Quality Manager Proxy	c:\winnt\system32\drivers\mspqm.sys	Kernel Driver	False					
	Demand Start	Stopped	OK	Normal	False	False			
mup	Mup	c:\winnt\system32\drivers\mup.sys	File System Driver		True	Boot Start			
	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 Start				
	Running	OK	Normal	False	True				
ndistapi	Remote Access NDIS TAPI Driver	c:\winnt\system32\drivers\ndistapi.sys	Kernel Driver	True					
	Demand Start	Running	OK	Normal	False	True			
ndiswan	Remote Access NDIS WAN Driver	c:\winnt\system32\drivers\ndiswan.sys	Kernel Driver	True					
	Demand Start	Running	OK	Normal	False	True			
ndproxy	NDIS Proxy	c:\winnt\system32\drivers\ndproxy.sys	Kernel Driver	True	Demand Start				
	Running	OK	Normal	False	True				
netbios	NetBIOS Interface	c:\winnt\system32\drivers\netbios.sys	File System Driver		True				
	System Start	Running	OK	Normal	False	True			
netbt	NetBios over Tcpip	c:\winnt\system32\drivers\netbt.sys	Kernel Driver	True	System Start				
	Running	OK	Normal	False	True				
netdetect	NetDetect	c:\winnt\system32\drivers\netdtect.sys	Kernel Driver	False	Demand Start				
	Stopped	OK	Normal	False	False				
npfs	Npfs	c:\winnt\system32\drivers\npfs.sys	File System Driver		True	System Start			
	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 Start	Running	OK		
	Normal	False	True						
nwlnkflt	IPX Traffic Filter Driver	c:\winnt\system32\drivers\nwlnkflt.sys	Kernel Driver	False	Demand Start				
	Stopped	OK	Normal	False	False				
nwlnkfwd	IPX Traffic Forwarder Driver	c:\winnt\system32\drivers\nwlnkfwd.sys	Kernel Driver	False					
	Demand Start	Stopped	OK	Normal	False	False			
parallel	Parallel class driver	c:\winnt\system32\drivers\parallel.sys	Kernel Driver	True	Demand Start				
	Running	OK	Normal	False	True				
parport	Parallel port driver	c:\winnt\system32\drivers\parport.sys	Kernel Driver	True	System Start				
	Running	OK	Ignore	False	True				
partmgr	PartMgr	c:\winnt\system32\drivers\partmgr.sys	Kernel Driver	True	Boot Start	Running	OK		
	Normal	False	True						
parvdm	ParVdm	c:\winnt\system32\drivers\parvdm.sys	Kernel Driver	True	Auto Start	Running	OK		
	Ignore	False	True						
pci	PCI Bus Driver	c:\winnt\system32\drivers\pci.sys	Kernel Driver	True	Boot Start				
	Running	OK	Critical	False	True				
pcidump	PCIDump	Not Available	Kernel Driver	False	System Start	Stopped	OK	Ignore	False
	False								
pciide	PCIIde	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
pcmcia	Pcmcia	c:\winnt\system32\drivers\pcmcia.sys	Kernel Driver	False	Disabled	Stopped	OK		
	Normal	False	False						
pdcomp	PDCOMP	Not Available	Kernel Driver	False	Demand Start		Stopped	OK	Ignore
	False	False							
pdframe	PDFRAME	Not Available	Kernel Driver	False	Demand Start		Stopped	OK	Ignore
	False	False							
pdreli	PDRELI	Not Available	Kernel Driver	False	Demand Start		Stopped	OK	Ignore
	False	False							
pdrframe	PDRFRAME	Not Available	Kernel Driver	False	Demand Start		Stopped	OK	Ignore
	False	False							
pptpminiport	WAN Miniport (PPTP)	c:\winnt\system32\drivers\raspptp.sys	Kernel Driver	True	Demand Start				
	Running	OK	Normal	False	True				
ptilink	Direct Parallel Link Driver	c:\winnt\system32\drivers\ptilink.sys	Kernel Driver	True	Demand Start				
	Running	OK	Normal	False	True				
ql1080	ql1080	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
ql10wnt	Ql10wnt	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
ql1240	ql1240	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
ql2100	ql2100	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
rasacd	Remote Access Auto Connection Driver	c:\winnt\system32\drivers\rasacd.sys	Kernel Driver	True					
	System Start	Running	OK	Normal	False	True			
rasl2tp	WAN Miniport (L2TP)	c:\winnt\system32\drivers\rasl2tp.sys	Kernel Driver	True	Demand Start				
	Running	OK	Normal	False	True				
raspti	Direct Parallel	c:\winnt\system32\drivers\raspti.sys	Kernel Driver	True	Demand Start				
	Running	OK	Normal	False	True				
rca	Microsoft Streaming Network Raw Channel Access	c:\winnt\system32\drivers\rca.sys	Kernel Driver	False					
	Demand Start	Stopped	OK	Normal	False	False			
rdbss	Rdbss	c:\winnt\system32\drivers\rdbss.sys	File System Driver		True	System Start			
	Running	OK	Normal	False	True				

APPENDIX C – TUNABLE PARAMETERS

rdpwd RDPWD c:\winnt\system32\drivers\rdpwd.sys Kernel Driver False Demand Start
Stopped OK Ignore False False
redbook Digital CD Audio Playback Filter Driver c:\winnt\system32\drivers\redbook.sys Kernel Driver False
System Start Stopped OK Normal False False
serenum Serenum Filter Driver c:\winnt\system32\drivers\serenum.sys Kernel Driver True Demand Start
Running OK Normal False True
serial Serial port driver c:\winnt\system32\drivers\serial.sys Kernel Driver True System Start
Running OK Ignore False True
sfloppy Sfloppy c:\winnt\system32\drivers\sfloppy.sys Kernel Driver False System Start Stopped OK
Ignore False False
sglfb sglfb Not Available Kernel Driver False System Start 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
Demand Start Running OK Normal False True
srv Srv c:\winnt\system32\drivers\srv.sys File System Driver True Demand Start
Running OK Normal False True
swenum Software Bus Driver c:\winnt\system32\drivers\swenum.sys Kernel Driver True Demand Start
Running OK Normal False True
symc810 symc810 Not Available Kernel Driver False Disabled Stopped OK Normal False
False
symc8xx symc8xx Not Available Kernel Driver False Disabled Stopped OK Normal False
False
sym_hi sym_hi Not Available Kernel Driver False Disabled Stopped OK Normal False
False
tcpip TCP/IP Protocol Driver c:\winnt\system32\drivers\tcpip.sys Kernel Driver True System Start
Running OK Normal False True
tdasync TDASYNC c:\winnt\system32\drivers\tdasync.sys Kernel Driver False Demand Start
Stopped OK Ignore False False
tdipx TDIPX c:\winnt\system32\drivers\tdipx.sys Kernel Driver False Demand Start
Stopped OK Ignore False False
tdnetb TDNETB c:\winnt\system32\drivers\tdnetb.sys Kernel Driver False Demand Start
Stopped OK Ignore False False
tdpipe TDPIPE c:\winnt\system32\drivers\tdpipe.sys Kernel Driver False Demand Start
Stopped OK Ignore False False
tdspx TDSPX c:\winnt\system32\drivers\tdspx.sys Kernel Driver False Demand Start
Stopped OK Ignore False False
tdtcp TDTCP c:\winnt\system32\drivers\tdtcp.sys Kernel Driver False Demand Start
Stopped OK Ignore False False
termdd Terminal Device Driver c:\winnt\system32\drivers\termdd.sys Kernel Driver False Disabled
Stopped OK Normal False False
tga tga Not Available Kernel Driver False System Start Stopped OK Ignore False
False
udfs Udfs c:\winnt\system32\drivers\udfs.sys File System Driver False Disabled
Stopped OK Normal False False
uhcd Microsoft USB Universal Host Controller Driver c:\winnt\system32\drivers\uhcd.sys Kernel Driver True
Demand Start Running OK Normal False True
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 Demand Start
Running OK Normal False True
usbhub Microsoft USB Standard Hub Driver c:\winnt\system32\drivers\usbhub.sys Kernel Driver True
Demand Start Running OK Normal False True
vgasave VgaSave c:\winnt\system32\drivers\vga.sys Kernel Driver True System Start Running OK
Ignore False True
wanarp Remote Access IP ARP Driver c:\winnt\system32\drivers\wanarp.sys Kernel Driver True
Demand Start Running OK Normal False True
wdica WDICA Not Available Kernel Driver False Demand Start Stopped OK Ignore
False False

[Environment Variables]

Variable Value User Name
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
Os2LibPath %SystemRoot%\system32\os2\dll; <SYSTEM>
Path %SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\Program Files\Microsoft SQL Server\80\Tools\BINN <SYSTEM>
windir %SystemRoot% <SYSTEM>
OS Windows_NT <SYSTEM>
PROCESSOR_ARCHITECTURE x86 <SYSTEM>
PROCESSOR_LEVEL 6 <SYSTEM>
PROCESSOR_IDENTIFIER x86 Family 6 Model 7 Stepping 3, GenuineIntel <SYSTEM>
PROCESSOR_REVISION 0703 <SYSTEM>
NUMBER_OF_PROCESSORS 1 <SYSTEM>
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH <SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>
TEMP %USERPROFILE%\Local Settings\Temp N1_CLIENT1\Administrator
TMP %USERPROFILE%\Local Settings\Temp N1_CLIENT1\Administrator
MSDevDir K:\Program Files\DevStudio\SharedIDE N1_CLIENT1\Administrator
path k\program files\devstudio\sharedide\bin\ide;k:\program files\devstudio\sharedide\bin;k:\program files\devstudio\vc\bin N1_CLIENT1\Administrator
lib k\program files\devstudio\vc\lib;k:\program files\devstudio\vc\mfc\lib;%lib% N1_CLIENT1\Administrator
include k\program files\devstudio\vc\include;k:\program files\devstudio\vc\atl\include;k:\program files\devstudio\vc\mfc\include;%include% N1_CLIENT1\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: \\n1_client5\c$ Disk Error
F: \\n1_client2\c$ Disk Error
H: \\n1_rte1\c$ Disk Error
I: \\n1_client3\c$ Disk Error
J: \\n1_client4\c$ Disk Error
K: \\prf_sut6\c$ Disk Error
L: \\nrddata\g$ Disk OK N1_CLIENT1\Administrator

[Running Tasks]

APPENDIX C – TUNABLE PARAMETERS

Name	Path	Process ID	Priority	Min Working Set		Max Working Set		Start Time
	Version	Size	File Date					
system idle process		Not Available	0	0	Not Available	Not Available	Not Available	Unknown
	Unknown	Unknown						
system	Not Available	8	8	0	1413120	Not Available	Unknown	Unknown
	Unknown							
smss.exe	c:\winnt\system32\smss.exe			160	11	204800	1413120	9/5/2000 12:26:37
PM	5.00.2090.1	42.77 KB (43,792 bytes)		Not Available				
csrss.exe	Not Available	184	13	Not Available	Not Available	9/5/2000 12:26:47 PM		Unknown
	Unknown	Unknown						
winlogon.exe	c:\winnt\system32\winlogon.exe			204	13	204800	1413120	9/5/2000 12:26:49
PM	5.00.2116.1	171.27 KB (175,376 bytes)		Not Available				
services.exe	c:\winnt\system32\services.exe			232	9	204800	1413120	9/5/2000 12:26:51
PM	5.00.2106.1	87.27 KB (89,360 bytes)		Not Available				
lsass.exe	c:\winnt\system32\lsass.exe			244	13	204800	1413120	9/5/2000 12:26:51
PM	5.00.2121.1	32.77 KB (33,552 bytes)		Not Available				
svchost.exe	c:\winnt\system32\svchost.exe			412	8	204800	1413120	9/5/2000 12:26:57
PM	5.00.2090.1	7.77 KB (7,952 bytes)		Not Available				
spoolsv.exe	c:\winnt\system32\spoolsv.exe			444	8	204800	1413120	9/5/2000 12:26:58
PM	5.00.2107.1	43.77 KB (44,816 bytes)		Not Available				
msdtc.exe	c:\winnt\system32\msdtc.exe			472	8	204800	1413120	9/5/2000 12:26:58
PM	1999.8.3413.3		6.77 KB (6,928 bytes)		Not Available			
svchost.exe	c:\winnt\system32\svchost.exe			580	8	204800	1413120	9/5/2000 12:27:03
PM	5.00.2090.1	7.77 KB (7,952 bytes)		Not Available				
llssrv.exe	c:\winnt\system32\llssrv.exe			604	9	204800	1413120	9/5/2000 12:27:03
PM	5.00.2090.1	113.77 KB (116,496 bytes)		Not Available				
regsvc.exe	c:\winnt\system32\regsvc.exe			660	8	204800	1413120	9/5/2000 12:27:03
PM	5.00.2091.1	63.77 KB (65,296 bytes)		Not Available				
mstask.exe	c:\winnt\system32\mstask.exe			708	8	204800	1413120	9/5/2000 12:27:05
PM	4.71.2113.1	114.77 KB (117,520 bytes)		Not Available				
tcpsvcs.exe	c:\winnt\system32\tcpsvcs.exe			748	8	204800	1413120	9/5/2000 12:27:06
PM	5.00.2090.1	24.77 KB (25,360 bytes)		Not Available				
inetinfo.exe	c:\winnt\system32\inetsrv\inetinfo.exe			852	8	204800	1413120	9/5/2000 12:27:09
PM	5.00.0984	14.27 KB (14,608 bytes)		Not Available				
dfssvc.exe	c:\winnt\system32\dfssvc.exe			876	8	204800	1413120	9/5/2000 12:27:10
PM	5.00.2124.1	95.77 KB (98,064 bytes)		Not Available				
explorer.exe	c:\winnt\explorer.exe		1028	8	204800	1413120	9/5/2000 12:27:34 PM	
	5.00.2919.3800		232.77 KB (238,352 bytes)		Not Available			
hpmon.exe	c:\winnt\system32\hpmon.exe			1080	8	204800	1413120	9/5/2000 12:27:38
PM	1.11	28.50 KB (29,184 bytes)		Not Available				
svchost.exe	c:\winnt\system32\svchost.exe			1144	8	204800	1413120	9/5/2000 12:27:52
PM	5.00.2090.1	7.77 KB (7,952 bytes)		Not Available				
mmc.exe	c:\winnt\system32\mmc.exe			504	8	204800	1413120	9/5/2000 12:54:24
PM	5.00.2115.1	589.27 KB (603,408 bytes)		Not Available				
winmgmt.exe	c:\winnt\system32\wbem\winmgmt.exe			1120	8	204800	1413120	9/5/2000 12:54:57
PM	1.50.1025.0015		164.05 KB (167,991 bytes)		Not Available			
rsvp.exe	c:\winnt\system32\rsvp.exe			696	8	204800	1413120	9/5/2000 12:55:31
PM	5.00.2120.1	170.77 KB (174,864 bytes)		Not Available				

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
traffic.dll	5.00.2090.1	30.77 KB (31,504 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\traffic.dll				
rsvp.exe	5.00.2120.1	170.77 KB (174,864 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\rsvp.exe				
rapilib.dll	5.00.2120.1	24.77 KB (25,360 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\rapilib.dll				
rsvpsp.dll	5.00.2120.1	74.77 KB (76,560 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\rsvpsp.dll				
provthrd.dll	1.50.1025.0001	68.08 KB (69,713 bytes)	11/9/1999 1:53:16 PM	Microsoft Corporation	
	c:\winnt\system32\wbem\provthrd.dll				
ntevt.dll	1.50.1025.0002	176.06 KB (180,290 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\wbem\ntevt.dll				
ntmarta.dll	5.00.2119.1	98.27 KB (100,624 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\ntmarta.dll				
perfos.dll	5.0	21.27 KB (21,776 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\perfos.dll				
psapi.dll	5.00.2090.1	28.27 KB (28,944 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\psapi.dll				
framedyn.dll	1.50.1025.0002	164.05 KB (167,988 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\wbem\framedyn.dll				
cimwin32.dll	1.50.1025.0016	1.02 MB (1,065,018 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\wbem\cimwin32.dll				
wbemess.dll	1.50.1025.0009	324.05 KB (331,827 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\wbem\wbemess.dll				
wbemcore.dll	1.50.1025.0012	592.05 KB (606,260 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\wbem\wbemcore.dll				
winmgmt.exe	1.50.1025.0015	164.05 KB (167,991 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\wbem\winmgmt.exe				
fastprox.dll	1.50.1025.0009	144.08 KB (147,536 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\wbem\fastprox.dll				
wbemsvc.dll	1.50.1025.0009	136.07 KB (139,339 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\wbem\wbemsvc.dll				
wbemcomn.dll	1.50.1025.0009	688.05 KB (704,564 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\wbem\wbemcomn.dll				
wbemprox.dll	1.50.1025.0009	40.05 KB (41,012 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\wbem\wbemprox.dll				
mlang.dll	5.00.2919.3800	509.77 KB (522,000 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\mlang.dll				
rassapi.dll	5.00.2116.1	14.27 KB (14,608 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\rassapi.dll				
adsnt.dll	5.00.2118.1	193.77 KB (198,416 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\adsnt.dll				
dbghelp.dll	5.00.2128.1	77.27 KB (79,120 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\dbghelp.dll				
localsec.dll	5.00.2099.1	226.77 KB (232,208 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\localsec.dll				
devmgr.dll	5.00.2109.1	215.27 KB (220,432 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\devmgr.dll				
filemgmt.dll	5.00.2116.1	287.27 KB (294,160 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\filemgmt.dll				
pdh.dll	5.00.1838.1	135.77 KB (139,024 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\pdh.dll				
smlogcfg.dll	5.00.2107.1	272.77 KB (279,312 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\smlogcfg.dll				
cabinet.dll	5.00.2090.1	54.77 KB (56,080 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\cabinet.dll				
msinfo32.dll	5.00.2121.1	306.27 KB (313,616 bytes)	11/9/1999 1:53:26 PM	Microsoft Corporation	c:\program
files\common files\microsoft shared\msinfo\msinfo32.dll					
riched20.dll	5.30.20.1200	419.77 KB (429,840 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\riched20.dll				

APPENDIX C – TUNABLE PARAMETERS

riched32.dll 5.00.2090.1 3.77 KB (3,856 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\riched32.dll
els.dll 5.00.2108.1 146.77 KB (150,288 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\els.dll
ntmsmgr.dll 1,0,0,1 427.27 KB (437,520 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation and HighGround Systems, Inc. c:\winnt\system32\ntmsmgr.dll
mmfutil.dll 1.50.1025.0005 32.06 KB (32,834 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\mmfutil.dll
logdrive.dll 1.50.1025.0004 200.07 KB (204,868 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\logdrive.dll
dfrgres.dll 5.00.2109.1 27.50 KB (28,160 bytes) 9/9/1999 5:00:00 PM Executive Software International, Inc. c:\winnt\system32\dfrgres.dll
dfrgsnap.dll 5.00.2109.1 41.77 KB (42,768 bytes) 9/9/1999 5:00:00 PM Executive Software International, Inc. c:\winnt\system32\dfrgsnap.dll
dmdskres.dll 2121.1.286.1 119.00 KB (121,856 bytes) 9/9/1999 5:00:00 PM Microsoft Corp., VERITAS Software c:\winnt\system32\dmdskres.dll
dmutil.dll 2121.1.286.1 41.77 KB (42,768 bytes) 9/9/1999 5:00:00 PM VERITAS Software Corp. c:\winnt\system32\dmutil.dll
ntmsapi.dll 5.00.1948.1 53.27 KB (54,544 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\ntmsapi.dll
dmdskmgr.dll 2121.1.286.1 158.27 KB (162,064 bytes) 9/9/1999 5:00:00 PM Microsoft Corp., VERITAS Software c:\winnt\system32\dmdskmgr.dll
mycomput.dll 5.00.2090.1 107.77 KB (110,352 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\mycomput.dll
mmcndmgr.dll 5.00.2108.1 815.27 KB (834,832 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\mmcndmgr.dll
mmc.exe 5.00.2115.1 589.27 KB (603,408 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\mmc.exe
tapisrv.dll 5.00.2114.1 170.77 KB (174,864 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\tapisrv.dll
hpmon.exe 1.11 28.50 KB (29,184 bytes) 5/22/2000 2:55:58 PM Hewlett-Packard Company c:\winnt\system32\hpmon.exe
wininet.dll 5.00.2919.3800 456.77 KB (467,728 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\wininet.dll
hhsetup.dll 4.74.8576 66.27 KB (67,856 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\hhsetup.dll
mmcshext.dll 5.00.2108.1 24.27 KB (24,848 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\mmcshext.dll
diskcopy.dll 5.00.2091.1 15.77 KB (16,144 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\diskcopy.dll
urlmon.dll 5.00.2919.3800 426.77 KB (437,008 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\urlmon.dll
faxshell.dll 5.00.2101.1 8.27 KB (8,464 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\faxshell.dll
msacm32.dll 5.00.2113.1 65.27 KB (66,832 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\msacm32.dll
avifil32.dll 5.00.2113.1 76.27 KB (78,096 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\avifil32.dll
msvfw32.dll 5.00.2113.1 113.77 KB (116,496 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\msvfw32.dll
docprop2.dll 5.00.2115.1 297.77 KB (304,912 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\docprop2.dll
linkinfo.dll 5.00.2091.1 15.77 KB (16,144 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\linkinfo.dll
browselc.dll 5.00.2919.3800 34.50 KB (35,328 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\browselc.dll
msi.dll 1.10.0816.3 1.64 MB (1,715,984 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\msi.dll
powrprof.dll 5.00.2919.3800 13.27 KB (13,584 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\powrprof.dll
batmeter.dll 5.00.2919.3800 20.27 KB (20,752 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\batmeter.dll
stobject.dll 5.00.2120.1 81.27 KB (83,216 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\stobject.dll
webcheck.dll 5.00.2919.3800 251.77 KB (257,808 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\webcheck.dll
ntshrui.dll 5.00.2090.1 46.77 KB (47,888 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\ntshrui.dll
mydocs.dll 5.00.2919.3800 55.77 KB (57,104 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\mydocs.dll
browseui.dll 5.00.2919.3800 791.77 KB (810,768 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\browseui.dll
shdocvw.dll 5.00.2919.3800 1.05 MB (1,103,632 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\shdocvw.dll
explorer.exe 5.00.2919.3800 232.77 KB (238,352 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\explorer.exe
dfssvc.exe 5.00.2124.1 95.77 KB (98,064 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\dfssvc.exe
iislog.dll 5.00.0984 75.77 KB (77,584 bytes) 11/8/1999 5:40:46 PM Microsoft Corporation c:\winnt\system32\inetsrv\iislog.dll
wshnetbs.dll 5.00.2090.1 7.77 KB (7,952 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\wshnetbs.dll
httpext.dll 0.9.3938.5 415.77 KB (425,744 bytes) 11/8/1999 5:40:46 PM Microsoft Corporation c:\winnt\system32\inetsrv\httpext.dll
rpcproxy.dll 5.00.2128.1 16.27 KB (16,656 bytes) 11/8/1999 5:39:47 PM Microsoft Corporation c:\winnt\system32\rpcproxy\rpcproxy.dll
fpexedll.dll 4.0.2.3228 20.06 KB (20,541 bytes) 11/8/1999 5:43:20 PM Microsoft Corporation c:\program files\common files\microsoft shared\web server extensions\40\bin\fpexedll.dll
md5filt.dll 5.00.0984 32.77 KB (33,552 bytes) 11/8/1999 5:40:53 PM Microsoft Corporation c:\winnt\system32\inetsrv\md5filt.dll
gzip.dll 5.00.0984 30.27 KB (30,992 bytes) 11/8/1999 5:40:52 PM Microsoft Corporation c:\winnt\system32\inetsrv\gzip.dll
compfilt.dll 5.00.0984 22.27 KB (22,800 bytes) 11/8/1999 5:40:52 PM Microsoft Corporation c:\winnt\system32\inetsrv\compfilt.dll
sspifilt.dll 5.00.0984 42.77 KB (43,792 bytes) 11/8/1999 5:40:54 PM Microsoft Corporation c:\winnt\system32\inetsrv\sspifilt.dll
iscomlog.dll 5.00.0984 24.27 KB (24,848 bytes) 11/8/1999 5:40:47 PM Microsoft Corporation c:\winnt\system32\inetsrv\iscomlog.dll
lonsint.dll 5.00.0984 11.77 KB (12,048 bytes) 11/8/1999 5:40:47 PM Microsoft Corporation c:\winnt\system32\inetsrv\lonsint.dll
inetsloc.dll 5.00.0984 20.27 KB (20,752 bytes) 11/8/1999 5:40:48 PM Microsoft Corporation c:\winnt\system32\inetsloc.dll
iisfecnv.dll 5.00.0984 7.27 KB (7,440 bytes) 11/8/1999 5:40:46 PM Microsoft Corporation c:\winnt\system32\inetsrv\iisfecnv.dll
isatq.dll 5.00.0984 60.27 KB (61,712 bytes) 11/8/1999 5:40:49 PM Microsoft Corporation c:\winnt\system32\inetsrv\isatq.dll
infocomm.dll 5.00.0984 230.27 KB (235,792 bytes) 11/8/1999 5:40:47 PM Microsoft Corporation c:\winnt\system32\inetsrv\infocomm.dll
w3svc.dll 5.00.0984 345.27 KB (353,552 bytes) 11/8/1999 5:40:54 PM Microsoft Corporation c:\winnt\system32\inetsrv\w3svc.dll
security.dll 5.00.2112.1 5.77 KB (5,904 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation c:\winnt\system32\security.dll

APPENDIX C – TUNABLE PARAMETERS

svcext.dll 5.00.0984 39.27 KB (40,208 bytes) 11/8/1999 5:40:47 PM Microsoft Corporation
c:\winnt\system32\inetsrv\svcext.dll
admexs.dll 5.00.0984 27.77 KB (28,432 bytes) 11/8/1999 5:40:46 PM Microsoft Corporation
c:\winnt\system32\inetsrv\admexs.dll
wamreg.dll 5.00.0984 44.77 KB (45,840 bytes) 11/8/1999 5:40:55 PM Microsoft Corporation
c:\winnt\system32\inetsrv\wamreg.dll
metadata.dll 5.00.0984 68.27 KB (69,904 bytes) 11/8/1999 5:40:47 PM Microsoft Corporation
c:\winnt\system32\inetsrv\metadata.dll
iismap.dll 5.00.0984 56.27 KB (57,616 bytes) 11/8/1999 5:40:48 PM Microsoft Corporation
c:\winnt\system32\iismap.dll
nsepm.dll 5.00.0984 43.27 KB (44,304 bytes) 11/8/1999 5:40:47 PM Microsoft Corporation
c:\winnt\system32\inetsrv\nsepm.dll
admwprox.dll 5.00.0984 31.27 KB (32,016 bytes) 11/8/1999 5:40:48 PM Microsoft Corporation
c:\winnt\system32\admwprox.dll
coadmin.dll 5.00.0984 39.27 KB (40,208 bytes) 11/8/1999 5:40:49 PM Microsoft Corporation
c:\winnt\system32\inetsrv\coadmin.dll
iisadmin.dll 5.00.0984 14.77 KB (15,120 bytes) 11/8/1999 5:40:46 PM Microsoft Corporation
c:\winnt\system32\inetsrv\iisadmin.dll
rpcref.dll 5.00.0984 4.27 KB (4,368 bytes) 11/8/1999 5:40:47 PM Microsoft Corporation
c:\winnt\system32\inetsrv\rpcref.dll
iisrtl.dll 5.00.0984 119.27 KB (122,128 bytes) 11/8/1999 5:40:48 PM Microsoft Corporation
c:\winnt\system32\iisrtl.dll
inetinfo.exe 5.00.0984 14.27 KB (14,608 bytes) 11/8/1999 5:40:47 PM Microsoft Corporation
c:\winnt\system32\inetsrv\inetinfo.exe
simptcp.dll 5.00.2106.1 19.27 KB (19,728 bytes) 11/8/1999 5:39:47 PM Microsoft Corporation
c:\winnt\system32\simptcp.dll
tcpsvcs.exe 5.00.2090.1 24.77 KB (25,360 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\tcpsvcs.exe
msidle.dll 5.00.2919.3800 6.27 KB (6,416 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\msidle.dll
mstask.exe 4.71.2113.1 114.77 KB (117,520 bytes) 11/9/1999 1:53:16 PM Microsoft Corporation
c:\winnt\system32\mstask.exe
regsvc.exe 5.00.2091.1 63.77 KB (65,296 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\regsvc.exe
llsrpc.dll 5.00.2107.1 45.77 KB (46,864 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\llsrpc.dll
llssrv.exe 5.00.2090.1 113.77 KB (116,496 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\llssrv.exe
ntmsdba.dll 5.00.2108.1 167.27 KB (171,280 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\ntmsdba.dll
rasdlg.dll 5.00.2120.1 512.77 KB (525,072 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\rasdlg.dll
netcfgx.dll 5.00.2120.1 532.27 KB (545,040 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\netcfgx.dll
rasmans.dll 5.00.2119.1 150.27 KB (153,872 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\rasmans.dll
wmi.dll 5.00.2112.1 6.27 KB (6,416 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\wmi.dll
netshell.dll 5.00.2120.1 453.77 KB (464,656 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\netshell.dll
netman.dll 5.00.2120.1 88.77 KB (90,896 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\netman.dll
iashlpr.dll 5.00.2090.1 31.27 KB (32,016 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\iashlpr.dll
iasacct.dll 5.00.2095.1 28.27 KB (28,944 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\iasacct.dll
iasuserr.dll 5.00.2090.1 25.77 KB (26,384 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\iasuserr.dll
iasnap.dll 5.00.2090.1 58.77 KB (60,176 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\iasnap.dll
iaspipe.dll 5.00.2090.1 41.77 KB (42,768 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\iaspipe.dll
expsrv.dll 6.0.8540 370.27 KB (379,152 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\expsrv.dll
vbajet32.dll 6.1.8268 30.27 KB (30,992 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\vbajet32.dll
msjtes40.dll 4.00.2927.8 232.27 KB (237,840 bytes) 3/13/2000 7:20:46 PM Microsoft Corporation
c:\winnt\system32\msjtes40.dll
oledb32r.dll 2.60.4815.0 68.27 KB (69,904 bytes) 3/13/2000 7:20:47 PM Microsoft Corporation c:\program files\common files\system\ole db\oledb32r.dll
msdart32.dll 2.60.4815.0 84.27 KB (86,288 bytes) 3/13/2000 7:20:45 PM Microsoft Corporation
c:\winnt\system32\msdart32.dll
oledb32.dll 2.60.4815.0 460.27 KB (471,312 bytes) 3/13/2000 7:20:47 PM Microsoft Corporation c:\program files\common files\system\ole db\oledb32.dll
msjint40.dll 4.00.2927.2 148.27 KB (151,824 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\msjint40.dll
msjter40.dll 4.00.2927.2 52.27 KB (53,520 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\msjter40.dll
mswstr10.dll 4.00.2927.10 600.27 KB (614,672 bytes) 3/13/2000 7:21:06 PM Microsoft Corporation
c:\winnt\system32\mswstr10.dll
msjet40.dll 4.00.2927.4 1.43 MB (1,495,312 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\msjet40.dll
msjetoledb40.dll 4.00.2927.2 340.27 KB (348,432 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\msjetoledb40.dll
iasrad.dll 5.00.2090.1 94.27 KB (96,528 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\iasrad.dll
iassam.dll 5.00.2090.1 96.27 KB (98,576 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\iassam.dll
iasads.dll 5.00.2112.1 73.77 KB (75,536 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\iasads.dll
sens.dll 5.00.2090.1 35.77 KB (36,624 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\sens.dll
iaspolcy.dll 5.00.2090.1 25.27 KB (25,872 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\iaspolcy.dll
iassvcs.dll 5.00.2090.1 58.27 KB (59,664 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\iassvcs.dll
iassdo.dll 5.00.2090.1 262.27 KB (268,560 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\iassdo.dll
ntmssvc.dll 5.00.2108.1 390.27 KB (399,632 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\ntmssvc.dll
ias.dll 5.00.2090.1 7.27 KB (7,440 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\ias.dll
es.dll 1999.8.3413.3 220.77 KB (226,064 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\es.dll
mtxoci.dll 1999.8.3413.3 101.77 KB (104,208 bytes) 11/8/1999 5:39:56 PM Microsoft Corporation
c:\winnt\system32\mtxoci.dll
resutils.dll 5.00.2123.1 39.77 KB (40,720 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\resutils.dll
clusapi.dll 5.00.2123.1 49.27 KB (50,448 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\clusapi.dll
msvcp50.dll 5.00.7051 552.50 KB (565,760 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\msvcp50.dll
xolehlp.dll 1999.8.3413.3 18.27 KB (18,704 bytes) 11/8/1999 5:39:55 PM Microsoft Corporation
c:\winnt\system32\xolehlp.dll
msdtclog.dll 1999.8.3413.3 85.27 KB (87,312 bytes) 11/8/1999 5:39:55 PM Microsoft Corporation
c:\winnt\system32\msdtclog.dll

APPENDIX C – TUNABLE PARAMETERS

mtxclu.dll	1999.8.3413.3	50.77 KB (51,984 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\mtxclu.dll
msdtcprx.dll	1999.8.3413.7	625.77 KB (640,784 bytes)	11/8/1999 5:39:56 PM	Microsoft Corporation	c:\winnt\system32\msdtcprx.dll
txfaux.dll	1999.8.3413.7	370.77 KB (379,664 bytes)	11/8/1999 5:39:55 PM	Microsoft Corporation	c:\winnt\system32\txfaux.dll
msdtctm.dll	1999.8.3413.7	1.07 MB (1,120,528 bytes)	11/8/1999 5:39:56 PM	Microsoft Corporation	c:\winnt\system32\msdtctm.dll
msdtc.exe	1999.8.3413.3	6.77 KB (6,928 bytes)	11/8/1999 5:39:55 PM	Microsoft Corporation	c:\winnt\system32\msdtc.exe
inetpp.dll	5.00.2090.1	62.77 KB (64,272 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\inetpp.dll
win32spl.dll	5.00.2092.1	81.27 KB (83,216 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\win32spl.dll
usbmon.dll	5.00.2116.1	11.27 KB (11,536 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\usbmon.dll
tcpmon.dll	5.00.2102.1	40.77 KB (41,744 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\tcpmon.dll
pjlmon.dll	5.00.2090.1	12.77 KB (13,072 bytes)	9/9/1999 3:37:34 AM	Microsoft Corporation	c:\winnt\system32\pjlmon.dll
cnbjmon.dll	5.00.2090.1	43.77 KB (44,816 bytes)	9/9/1999 3:37:26 AM	Microsoft Corporation	c:\winnt\system32\cnbjmon.dll
localspl.dll	5.00.2119.1	245.77 KB (251,664 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\localspl.dll
spoolss.dll	5.00.2110.1	60.77 KB (62,224 bytes)	11/8/1999 4:28:25 PM	Microsoft Corporation	c:\winnt\system32\spoolss.dll
spoolsv.exe	5.00.2107.1	43.77 KB (44,816 bytes)	11/8/1999 4:28:25 PM	Microsoft Corporation	c:\winnt\system32\spoolsv.exe
rasadhlp.dll	5.00.2109.1	7.27 KB (7,440 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\rasadhlp.dll
winrnr.dll	5.00.2110.1	18.77 KB (19,216 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\winrnr.dll
rpcss.dll	5.00.2119.1	225.27 KB (230,672 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\rpcss.dll
svchost.exe	5.00.2090.1	7.77 KB (7,952 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\svchost.exe
dssbase.dll	5.00.2120.1	140.27 KB (143,632 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\dssbase.dll
wshtcpip.dll	5.00.2090.1	17.27 KB (17,680 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\wshtcpip.dll
msafd.dll	5.00.2095.1	52.27 KB (53,520 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\msafd.dll
oakley.dll	5.00.2115.1	420.27 KB (430,352 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\oakley.dll
mfc42u.dll	6.00.8576.0	972.05 KB (995,384 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\mfc42u.dll
polagent.dll	5.00.2110.1	102.27 KB (104,720 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\polagent.dll
scecli.dll	5.00.2112.1	101.77 KB (104,208 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\scecli.dll
esent.dll	6.0.3938.7	848.77 KB (869,136 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\esent.dll
mswsock.dll	5.00.2120.1	62.77 KB (64,272 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\mswsock.dll
ntdsatq.dll	5.00.2122.1	30.77 KB (31,504 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\ntdsatq.dll
ntdsa.dll	5.00.2127.1	984.77 KB (1,008,400 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\ntdsa.dll
kdcsvc.dll	5.00.2121.1	138.77 KB (142,096 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\kdcsvc.dll
sfmapi.dll	5.00.2090.1	38.77 KB (39,696 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\sfmapi.dll
rassfm.dll	5.00.2109.1	21.27 KB (21,776 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\rassfm.dll
schannel.dll	5.00.2118.1	136.77 KB (140,048 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\schannel.dll
netlogon.dll	5.00.2119.1	344.77 KB (353,040 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\netlogon.dll
msv1_0.dll	5.00.2113.1	93.77 KB (96,016 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\msv1_0.dll
kerberos.dll	5.00.2121.1	190.27 KB (194,832 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\kerberos.dll
msprivs.dll	5.00.2112.1	41.50 KB (42,496 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\msprivs.dll
samsrv.dll	5.00.2124.1	352.27 KB (360,720 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\samsrv.dll
cryptdll.dll	5.00.2112.1	40.27 KB (41,232 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\cryptdll.dll
lsasrv.dll	5.00.2121.1	483.77 KB (495,376 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\lsasrv.dll
lsass.exe	5.00.2121.1	32.77 KB (33,552 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\lsass.exe
ntlsapi.dll	5.00.2090.1	6.77 KB (6,928 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\ntlsapi.dll
wmicore.dll	5.00.2119.1	70.27 KB (71,952 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\wmicore.dll
rnr20.dll	5.00.2120.1	35.27 KB (36,112 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\rnr20.dll
browser.dll	5.00.2098.1	48.27 KB (49,424 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\browser.dll
alrsvc.dll	5.00.2090.1	17.77 KB (18,192 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\alrsvc.dll
msgsvc.dll	5.00.2110.1	33.77 KB (34,576 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\msgsvc.dll
trkwks.dll	5.00.2110.1	87.77 KB (89,872 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\trkwks.dll
seclogon.dll	5.00.2122.1	15.27 KB (15,632 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\seclogon.dll
psbase.dll	5.00.2090.1	110.77 KB (113,424 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\psbase.dll
cryptsvc.dll	5.00.2090.1	66.77 KB (68,368 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\cryptsvc.dll
wkssvc.dll	5.00.2120.1	91.27 KB (93,456 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\wkssvc.dll
srvsvc.dll	5.00.2117.1	79.27 KB (81,168 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\srvsvc.dll
cfgmgr32.dll	5.00.2098.1	16.77 KB (17,168 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\cfgmgr32.dll
dmserver.dll	2121.1.286.1	11.77 KB (12,048 bytes)	9/9/1999 5:00:00 PM	VERITAS Software Corp.	c:\winnt\system32\dmserver.dll
winsta.dll	5.00.2100.1	36.27 KB (37,136 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\winsta.dll
dhcpcsvc.dll	5.00.2107.1	88.77 KB (90,896 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\dhcpcsvc.dll
tapi32.dll	5.00.2090.1	122.27 KB (125,200 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation	c:\winnt\system32\tapi32.dll

APPENDIX C – TUNABLE PARAMETERS

rasman.dll 5.00.2114.1 59.77 KB (61,200 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\rasman.dll
rasapi32.dll 5.00.2116.1 187.27 KB (191,760 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\rasapi32.dll
rtutils.dll 5.00.2109.1 43.27 KB (44,304 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\rtutils.dll
adsldpc.dll 5.00.2120.1 125.77 KB (128,784 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\adsldpc.dll
activeds.dll 5.00.2118.1 171.77 KB (175,888 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\activeds.dll
mprapi.dll 5.00.2112.1 90.77 KB (92,944 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\mprapi.dll
iphlpapi.dll 5.00.2095.2 67.27 KB (68,880 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\iphlpapi.dll
icmp.dll 5.00.2090.1 7.27 KB (7,440 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\icmp.dll
lmhsvc.dll 5.00.2102.1 9.27 KB (9,488 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\lmhsvc.dll
dnsrslvr.dll 5.00.2118.1 91.27 KB (93,456 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\dnsrslvr.dll
eventlog.dll 5.00.2090.1 43.77 KB (44,816 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\eventlog.dll
ntdsapi.dll 5.00.2120.1 55.27 KB (56,592 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\ntdsapi.dll
scesrv.dll 5.00.2112.1 220.27 KB (225,552 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\scesrv.dll
umpnpmgr.dll 5.00.2109.1 116.77 KB (119,568 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\umpnpmgr.dll
services.exe 5.00.2106.1 87.27 KB (89,360 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\services.exe
clbcatq.dll 1999.8.3413.3 494.27 KB (506,128 bytes) 11/8/1999 5:39:47 PM Microsoft Corporation
c:\winnt\system32\clbcatq.dll
oleaut32.dll 2.40.4505 596.27 KB (610,576 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\oleaut32.dll
netmsg.dll 5.00.2090.1 152.50 KB (156,160 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\netmsg.dll
comdlg32.dll 5.00.2919.3800 235.27 KB (240,912 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\comdlg32.dll
netui2.dll 5.00.2107.1 280.27 KB (286,992 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\netui2.dll
mprui.dll 5.00.2090.1 54.77 KB (56,080 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\mprui.dll
netui1.dll 5.00.2107.1 209.77 KB (214,800 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\netui1.dll
netui0.dll 5.00.2107.1 70.27 KB (71,952 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\netui0.dll
ntlanman.dll 5.00.2109.1 35.27 KB (36,112 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\ntlanman.dll
mpr.dll 5.00.2111.1 53.27 KB (54,544 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\mpr.dll
cscui.dll 5.00.2116.1 225.77 KB (231,184 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\cscui.dll
atl.dll 3.00.8449 57.56 KB (58,938 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\atl.dll
certcli.dll 5.00.2120.1 131.27 KB (134,416 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\certcli.dll
winspool.drv 5.00.2110.1 109.77 KB (112,400 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\winspool.drv

winscard.dll 5.00.2108.1 77.27 KB (79,120 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\winscard.dll
winmm.dll 5.00.2114.1 184.27 KB (188,688 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\winmm.dll
wlnotify.dll 5.00.2090.1 52.77 KB (54,032 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\wlnotify.dll
cscdll.dll 5.00.2122.1 97.77 KB (100,112 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\cscdll.dll
lz32.dll 5.00.2090.1 9.77 KB (10,000 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\lz32.dll
version.dll 5.00.2090.1 15.77 KB (16,144 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\version.dll
rsabase.dll 5.00.2120.1 127.77 KB (130,832 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\rsabase.dll
setupapi.dll 5.00.2126.1 551.27 KB (564,496 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\setupapi.dll
mscat32.dll 5.131.2090.1 7.77 KB (7,952 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\mscat32.dll
ole32.dll 5.00.2120.1 961.27 KB (984,336 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\ole32.dll
imagehlp.dll 5.00.2128.1 40.77 KB (41,744 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\imagehlp.dll
msasn1.dll 5.00.2090.1 50.27 KB (51,472 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\msasn1.dll
crypt32.dll 5.131.2118.1 454.77 KB (465,680 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\crypt32.dll
wintrust.dll 5.131.2090.1 161.27 KB (165,136 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\wintrust.dll
comctl32.dll 5.81 539.77 KB (552,720 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\comctl32.dll
shlwapi.dll 5.00.2919.3800 281.77 KB (288,528 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\shlwapi.dll
shell32.dll 5.00.2919.3800 2.24 MB (2,344,208 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\shell32.dll
msgina.dll 5.00.2115.1 308.27 KB (315,664 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\msgina.dll
wsock32.dll 5.00.2120.1 21.27 KB (21,776 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\wsock32.dll
dnsapi.dll 5.00.2120.1 133.77 KB (136,976 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\dnsapi.dll
wldap32.dll 5.00.2117.1 153.77 KB (157,456 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\wldap32.dll
ws2help.dll 5.00.2095.1 17.77 KB (18,192 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\ws2help.dll
ws2_32.dll 5.00.2104.1 67.77 KB (69,392 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\ws2_32.dll
samlib.dll 5.00.2124.1 46.27 KB (47,376 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\samlib.dll
netrap.dll 5.00.2090.1 11.27 KB (11,536 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\netrap.dll
netapi32.dll 5.00.2120.1 295.77 KB (302,864 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\netapi32.dll
profmap.dll 5.00.2112.1 27.27 KB (27,920 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\profmap.dll
secur32.dll 5.00.2119.1 44.77 KB (45,840 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\secur32.dll
sfc.dll 5.00.2124.1 83.27 KB (85,264 bytes) 9/9/1999 5:00:00 PM Microsoft Corporation
c:\winnt\system32\sfc.dll

APPENDIX C – TUNABLE PARAMETERS

nddeapi.dll	5.00.2090.1	15.27 KB (15,632 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation
	c:\winnt\system32\nddeapi.dll			
userenv.dll	5.00.2127.1	343.27 KB (351,504 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation
	c:\winnt\system32\userenv.dll			
user32.dll	5.00.2120.1	392.27 KB (401,680 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation
	c:\winnt\system32\user32.dll			
gdi32.dll	5.00.2115.1	228.27 KB (233,744 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation
	c:\winnt\system32\gdi32.dll			
rpcrt4.dll	5.00.2128.1	440.27 KB (450,832 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation
	c:\winnt\system32\rpcrt4.dll			
advapi32.dll	5.00.2120.1	337.77 KB (345,872 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation
	c:\winnt\system32\advapi32.dll			
kernel32.dll	5.00.2122.1	711.77 KB (728,848 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation
	c:\winnt\system32\kernel32.dll			
msvcrt.dll	6.10.8581.0	284.05 KB (290,869 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation
	c:\winnt\system32\msvcrt.dll			
winlogon.exe	5.00.2116.1	171.27 KB (175,376 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation
	c:\winnt\system32\winlogon.exe			
sfcfiles.dll	5.00.2128.1	366.77 KB (375,568 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation
	c:\winnt\system32\sfcfiles.dll			
ntdll.dll	5.00.2121.1	469.27 KB (480,528 bytes)	9/9/1999 5:00:00 PM	Microsoft Corporation
	c:\winnt\system32\ntdll.dll			
smss.exe	5.00.2090.1	42.77 KB (43,792 bytes)	9/9/1999 5:00:00 PM	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 Start	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
Application Management	AppMgmt	Stopped	Demand Start	Share Process				
	c:\winnt\system32\services.exe		Normal	LocalSystem	0			
Computer Browser	Browser	Running	Auto Start	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
Indexing Service	cisvc	Stopped	Demand Start	Share Process				
	c:\winnt\system32\cisvc.exe		Normal	LocalSystem	0			
ClipBook	ClipSrv	Stopped	Demand Start	Own Process	c:\winnt\system32\clipsrv.exe			
	Normal	LocalSystem	0					
Distributed File System	Dfs	Running	Auto Start	Own Process	c:\winnt\system32\dfssvc.exe			
	Normal	LocalSystem	0					
DHCP Client	Dhcp	Stopped	Disabled	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
Logical Disk Manager Administrative Service			dmadmin	Stopped	Demand Start	Share Process		
	c:\winnt\system32\dmadmin.exe /com		Normal	LocalSystem	0			
Logical Disk Manager	dmserver	Running	Auto Start	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
DNS Client	Dnscache	Running	Auto Start	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
Event Log	Eventlog	Running	Auto Start	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
COM+ Event System	EventSystem	Running	Demand Start	Share Process				
	c:\winnt\system32\svchost.exe -k netsvcs			Normal	LocalSystem	0		
Fax Service	Fax	Stopped	Demand Start	Own Process	c:\winnt\system32\faxsvc.exe			
	Normal	LocalSystem	0					
Internet Authentication Service		IAS	Running	Auto Start	Share Process			
	c:\winnt\system32\svchost.exe -k netsvcs			Normal	LocalSystem	0		
IIS Admin Service		IISADMIN	Running	Auto Start	Share Process			
	c:\winnt\system32\inetsrv\inetinfo.exe			Normal	LocalSystem	0		
IMDB Server	ImdbServer	Stopped	Disabled	Own Process	c:\winnt\system32\imdbsrv.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 Start	Share Process	c:\winnt\system32\services.exe		
	Normal	LocalSystem	0					
Workstation	lanmanworkstation		Running	Auto Start	Share Process	c:\winnt\system32\services.exe		
	Normal	LocalSystem	0					
License Logging Service	LicenseService		Running	Auto Start	Own Process	c:\winnt\system32\llssrv.exe		
	Normal	LocalSystem	0					
TCP/IP NetBIOS Helper Service		LmHosts	Running	Auto Start	Share Process			
	c:\winnt\system32\services.exe			Normal	LocalSystem	0		
Messenger	Messenger	Running	Auto Start	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
NetMeeting Remote Desktop Sharing		mnmsrvc	Stopped	Demand Start	Own Process			
	c:\winnt\system32\mnmsrvc.exe			Normal	LocalSystem	0		
Distributed Transaction Coordinator		MSDTC	Running	Auto Start	Own Process	c:\winnt\system32\msdtc.exe		
	Normal	LocalSystem	0					
Windows Installer		MSIServer	Stopped	Demand Start	Share Process			
	c:\winnt\system32\msiexec.exe /v			Normal	LocalSystem	0		
Network DDE		NetDDE	Stopped	Demand Start	Share Process			
	c:\winnt\system32\netdde.exe			Normal	LocalSystem	0		
Network DDE DSDM		NetDDEdsdm		Stopped	Demand Start	Share Process		
	c:\winnt\system32\netdde.exe			Normal	LocalSystem	0		
Net Logon	Netlogon	Stopped	Demand Start		Share Process	c:\winnt\system32\lsass.exe		
	Normal	LocalSystem	0					
Network Connections	Netman	Running	Demand Start		Share Process			
	c:\winnt\system32\svchost.exe -k netsvcs			Normal	LocalSystem	0		
File Replication	NtFrs	Stopped	Demand Start		Own Process	c:\winnt\system32\ntfrs.exe		
	Ignore	LocalSystem	0					
NT LM Security Support Provider		NtLmSsp	Stopped	Demand Start	Share Process			
	c:\winnt\system32\lsass.exe			Normal	LocalSystem	0		
Removable Storage	NtmsSvc	Running	Auto Start	Share Process	c:\winnt\system32\svchost.exe -k			
netsvcs	Normal	LocalSystem	0					
Plug and Play		PlugPlay	Running	Auto Start	Share Process	c:\winnt\system32\services.exe		
	Normal	LocalSystem	0					
IPSEC Policy Agent		PolicyAgent	Running	Auto Start	Share Process	c:\winnt\system32\lsass.exe		
	Normal	LocalSystem	0					
Protected Storage		ProtectedStorage		Running	Auto Start	Share Process		
	c:\winnt\system32\services.exe			Normal	LocalSystem	0		
Remote Access Auto Connection Manager			RasAuto	Stopped	Demand Start	Share Process		
	c:\winnt\system32\svchost.exe -k netsvcs			Normal	LocalSystem	0		
Remote Access Connection Manager		RasMan	Stopped	Demand Start	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 Start	Own Process	c:\winnt\system32\regsvc.exe		
	Normal	LocalSystem	0					
Remote Procedure Call (RPC) Locator		RpcLocator	Stopped	Demand Start	Own Process			
	c:\winnt\system32\locator.exe			Normal	LocalSystem	0		
Remote Procedure Call (RPC)		RpcSs	Running	Auto Start	Share Process			
	c:\winnt\system32\svchost -k rpcss			Normal	LocalSystem	0		

APPENDIX C – TUNABLE PARAMETERS

```
QoS Admission Control (RSVP)  RSVP  Running  Auto Start  Own Process  c:\winnt\system32\rsvp.exe -s
        Normal  LocalSystem  0
Security Accounts Manager  SamSs  Running  Auto Start  Share Process
        c:\winnt\system32\lsass.exe  Normal  LocalSystem  0
Smart Card Helper  SCardDrv  Stopped  Demand Start  Share Process
        c:\winnt\system32\scardsvr.exe  Ignore  LocalSystem  0
Smart Card  SCardSvr  Stopped  Demand Start  Share Process  c:\winnt\system32\scardsvr.exe
        Ignore  LocalSystem  0
Task Scheduler  Schedule  Running  Auto Start  Share Process  c:\winnt\system32\mstask.exe
        Normal  LocalSystem  0
RunAs Service  seclogon  Running  Auto Start  Share Process  c:\winnt\system32\services.exe
        Ignore  LocalSystem  0
System Event Notification  SENS  Running  Auto Start  Share Process  c:\winnt\system32\svchost.exe -k
netsvcs  Normal  LocalSystem  0
Internet Connection Sharing  SharedAccess  Stopped  Demand Start  Share Process
        c:\winnt\system32\svchost.exe -k netsvcs  Normal  LocalSystem  0
Simple TCP/IP Services  SimpTcp  Running  Auto Start  Share Process  c:\winnt\system32\tcpsvcs.exe
        Normal  LocalSystem  0
Print Spooler  Spooler  Running  Auto Start  Own Process  c:\winnt\system32\spoolsv.exe  Normal
        LocalSystem  0
Performance Logs and Alerts  SysmonLog  Stopped  Auto Start  Own Process  c:\winnt\system32\smlogsvc.exe
        Normal  LocalSystem  0
Telephony  TapiSrv  Running  Demand Start  Share Process  c:\winnt\system32\svchost.exe -k
tapisrv  Normal  LocalSystem  0
Terminal Services  TermService  Stopped  Disabled  Own Process  c:\winnt\system32\termsrv.exe
        Normal  LocalSystem  0
Telnet  TlntSvr  Stopped  Demand Start  Own Process  c:\winnt\system32\tlntsvr.exe
        Normal  LocalSystem  0
Distributed Link Tracking Server  TrkSvr  Stopped  Demand Start  Share Process
        c:\winnt\system32\services.exe  Normal  LocalSystem  0
Distributed Link Tracking Client  TrkWks  Running  Auto Start  Share Process
        c:\winnt\system32\services.exe  Normal  LocalSystem  0
Uninterruptible Power Supply  UPS  Stopped  Demand Start  Own Process
        c:\winnt\system32\ups.exe  Normal  LocalSystem  0
Utility Manager  UtilMan  Stopped  Demand Start  Own Process  c:\winnt\system32\utilman.exe
        Normal  LocalSystem  0
Windows Time  W32Time  Stopped  Demand Start  Share Process
        c:\winnt\system32\services.exe  Normal  LocalSystem  0
World Wide Web Publishing Service  W3SVC  Running  Auto Start  Share Process
        c:\winnt\system32\inetsrv\inetinfo.exe  Normal  LocalSystem  0
Windows Management Instrumentation  WinMgmt  Running  Demand Start  Own Process
        c:\winnt\system32\wbem\winmgmt.exe  Ignore  LocalSystem  0
Windows Management Instrumentation Driver Extensions  Wmi  Running  Demand Start  Share
Process  c:\winnt\system32\services.exe  Normal  LocalSystem  0

[Program Groups]

Group Name  Name  User Name
Accessories  Default User:Accessories  Default User
Accessories\Accessibility  Default User:Accessories\Accessibility  Default User
Accessories\Entertainment  Default User:Accessories\Entertainment  Default User
Accessories\System Tools  Default User:Accessories\System Tools  Default User
Startup  Default User:Startup  Default User
Accessories  All Users:Accessories  All Users
Accessories\Accessibility  All Users:Accessories\Accessibility  All Users
Accessories\Communications  All Users:Accessories\Communications  All Users
Accessories\Entertainment  All Users:Accessories\Entertainment  All Users
Accessories\Games  All Users:Accessories\Games  All Users
Accessories\Microsoft Script Debugger  All Users:Accessories\Microsoft Script Debugger  All Users
Accessories\System Tools  All Users:Accessories\System Tools  All Users
Administrative Tools  All Users:Administrative Tools  All Users
Microsoft SQL Server  All Users:Microsoft SQL Server  All Users
Startup  All Users:Startup  All Users
Accessories  N1_CLIENT1\Administrator:Accessories  N1_CLIENT1\Administrator
Accessories\Accessibility  N1_CLIENT1\Administrator:Accessories\Accessibility  N1_CLIENT1\Administrator
Accessories\Communications  N1_CLIENT1\Administrator:Accessories\Communications
        N1_CLIENT1\Administrator
Accessories\Communications\HyperTerminal  N1_CLIENT1\Administrator:Accessories\Communications\HyperTerminal
        N1_CLIENT1\Administrator
Accessories\Entertainment  N1_CLIENT1\Administrator:Accessories\Entertainment  N1_CLIENT1\Administrator
Accessories\System Tools  N1_CLIENT1\Administrator:Accessories\System Tools  N1_CLIENT1\Administrator
Administrative Tools  N1_CLIENT1\Administrator:Administrative Tools  N1_CLIENT1\Administrator
Startup  N1_CLIENT1\Administrator:Startup  N1_CLIENT1\Administrator

[Startup Programs]

Program  Command  User Name  Location
mdac_runonce  c:\winnt\system32\runonce.exe  All Users
        HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HPmon.exe  hpmon.exe  All Users  HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

[OLE Registration]

Object  Local Server
Sound (OLE2)  sndrec32.exe
Media Clip  mplay32.exe
Video Clip  mplay32.exe /avi
MIDI Sequence  mplay32.exe /mid
Sound  Not Available
Media Clip  Not Available
Image Document  "C:\Program Files\Windows NT\Accessories\ImageVue\KodakImg.exe"
WordPad Document  "%ProgramFiles%\Windows NT\Accessories\WORDPAD.EXE"
Windows Media Services DRM Storage object  Not Available
Bitmap Image  C:\WINNT\System32\mspaint.exe

[Internet Explorer 5]

[ Following are sub-categories of this main category ]

[Summary]

Item  Value
Version  5.00.2919.3800
Build  52919.3800
Product ID  50293-270-1073316-10081
Application Path  C:\Program Files\Internet Explorer
Language  English (United States)
Active Printer  Not Available

Cipher Strength  56-bit
Content Advisor  Disabled
IEAK Install  No

[File Versions]
```

APPENDIX C – TUNABLE PARAMETERS

File	Version	Size	Date	Path	Company
advapi32.dll	5.0.2120.1	338 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
advpack.dll	5.0.2919.3800	87 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
browselc.dll	5.0.2919.3800	35 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
browseui.dll	5.0.2919.3800	792 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
ckcnv.exe	5.0.2120.1	9 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
comctl32.dll	5.81.2919.3800	540 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
crypt32.dll	5.131.2118.1	455 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
enhsig.dll	<File Missing>		Not Available	Not Available	Not Available	Not Available
iemigrat.dll	<File Missing>		Not Available	Not Available	Not Available	Not Available
iesetup.dll	5.0.2919.3800	57 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
iexplore.exe	5.0.2919.3800	59 KB	9/9/1999 5:00:00 PM	C:\Program Files\Internet Explorer	Microsoft Corporation
imagehlp.dll	5.0.2128.1	41 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
imghelp.dll	<File Missing>		Not Available	Not Available	Not Available	Not Available
inseng.dll	5.0.2919.3800	71 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
jobexec.dll	5.0.0.1	47 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
jscript.dll	5.1.0.4411	476 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
jsproxy.dll	5.0.2919.3800	13 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
msaahtml.dll	<File Missing>		Not Available	Not Available	Not Available	Not Available
mshtml.dll	5.0.2919.3800	2301 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
msjava.dll	5.0.3229.0	918 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
msoss.dll	<File Missing>		Not Available	Not Available	Not Available	Not Available
msxml.dll	5.0.2919.3800	509 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
occache.dll	5.0.2919.3800	86 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
ole32.dll	5.0.2120.1	961 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
oleaut32.dll	2.40.4505.1	596 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
olepro32.dll	5.0.4505.1	156 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
rsabase.dll	5.0.2120.1	128 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
rsaenh.dll	<File Missing>		Not Available	Not Available	Not Available	Not Available
rsapi32.dll	<File Missing>		Not Available	Not Available	Not Available	Not Available
rsasig.dll	<File Missing>		Not Available	Not Available	Not Available	Not Available
schannel.dll	5.0.2118.0	137 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
shdoc401.dll	<File Missing>		Not Available	Not Available	Not Available	Not Available
shdocvw.dll	5.0.2919.3800	1078 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
shell32.dll	5.0.2919.3800	2289 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
shlwapi.dll	5.0.2919.3800	282 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
url.dll	5.0.2919.3800	82 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
urlmon.dll	5.0.2919.3800	427 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
vbscript.dll	5.1.0.4411	428 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
webcheck.dll	5.0.2919.3800	252 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
win.com	5.0.2090.1	24 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
wininet.dll	5.0.2919.3800	457 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
winsock.dll	3.10.0.103	3 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
wintrust.dll	5.131.2090.1	161 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
wsock.vxd	<File Missing>		Not Available	Not Available	Not Available	Not Available
wsock32.dll	5.0.2120.1	21 KB	9/9/1999 5:00:00 PM	C:\WINNT\system32	Microsoft Corporation
wsock32n.dll	<File Missing>		Not Available	Not Available	Not Available	Not Available

[Connectivity]

Item	Value
Connection Preference	Never dial
EnableHttp1.1	1
ProxyHttp1.1	0

LAN Settings

AutoConfigProxy	wininet.dll
AutoProxyDetectMode	Enabled
AutoConfigURL	
Proxy	Disabled
ProxyServer	nsd-prxy.cup.hp.com:8088
ProxyOverride	<local>

[Cache]

[Following are sub-categories of this main category]

[Summary]

Item	Value
Page Refresh Type	Automatic
Temporary Internet Files Folder	C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files
Total Disk Space	8667 MB
Available Disk Space	3681 MB
Maximum Cache Size	270 MB
Available Cache Size	266 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]

APPENDIX C – TUNABLE PARAMETERS

Issued To Issued By Validity Signature Algorithm
Administrator Administrator 11/9/1999 to 10/15/2099 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.10 RTE Input Parameters

Profile: 3780_5cl_14dr
File Path: C:\benchcrf\3780_5cl_14dr.pro
Version: 1.0.1

Number of Engines: 14

Name: DRIVER1
Description: rte1_cpu0
Directory: c:\log\n1_rte10.log
Machine: n1_rte1
Parameter Set: PARAM9
Index: 757438
Seed: 28814
Configured Users: 2700
Pipe Name: DRIVER1-1672288046
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER10
Description: rte5_cpu1
Directory: c:\log\n1_rte51.log
Machine: n1_rte5
Parameter Set: PARAM9
Index: 900000000
Seed: 28814
Configured Users: 2700
Pipe Name: DRIVER10156751796
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER11
Description: rte6_cpu0
Directory: c:\log\n1_rte60.log
Machine: n1_rte6
Parameter Set: PARAM9
Index: 1000000000
Seed: 28814
Configured Users: 2700
Pipe Name: DRIVER11156849031
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER12
Description: rte6_cpu1
Directory: c:\log\n1_rte61.log
Machine: n1_rte6
Parameter Set: PARAM9
Index: 1100000000
Seed: 28814
Configured Users: 2700
Pipe Name: DRIVER12156883796
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER13
Description: rte7_cpu0
Directory: c:\log\n1_rte70.log
Machine: n1_rte7
Parameter Set: PARAM9
Index: 1200000000
Seed: 28814
Configured Users: 2700
Pipe Name: DRIVER13156926031
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER14
Description: rte7_cpu1
Directory: c:\log\n1_rte71.log
Machine: n1_rte7
Parameter Set: PARAM9
Index: 1300000000
Seed: 28814
Configured Users: 2700
Pipe Name: DRIVER14156976375
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

APPENDIX C – TUNABLE PARAMETERS

Name: DRIVER2
Description: rte1_cpu1
Directory: c:\log\n1_rte11.log
Machine: n1_rte1
Parameter Set: PARAM9
Index: 100000000
Seed: 28814
Configured Users: 2700
Pipe Name: DRIVER2-1672094593
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER3
Description: rte2_cpu0
Directory: c:\log\n1_rte20.log
Machine: n1_rte2
Parameter Set: PARAM9
Index: 348392290
Seed: 28814
Configured Users: 2700
Pipe Name: DRIVER3156231531
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER4
Description: rte2_cpu1
Directory: c:\log\n1_rte21.log
Machine: n1_rte2
Parameter Set: PARAM9
Index: 300000000
Seed: 28814
Configured Users: 2700
Pipe Name: DRIVER4156417500
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER5
Description: rte3_cpu0
Directory: c:\log\n1_rte30.log
Machine: n1_rte3
Parameter Set: PARAM9
Index: 400000000
Seed: 28814
Configured Users: 2700
Pipe Name: DRIVER5156476640
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER6
Description: rte3_cpu1
Directory: c:\log\n1_rte31.log
Machine: n1_rte3
Parameter Set: PARAM9
Index: 500000000
Seed: 28814
Configured Users: 2700
Pipe Name: DRIVER6156525906
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER7
Description: rte4_cpu0
Directory: c:\log\n1_rte40.log
Machine: n1_rte4
Parameter Set: PARAM9
Index: 600000000
Seed: 28814
Configured Users: 2700
Pipe Name: DRIVER7156613875
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER8
Description: rte4_cpu1
Directory: c:\log\n1_rte41.log
Machine: n1_rte4
Parameter Set: PARAM9
Index: 700000000
Seed: 28814
Configured Users: 2700
Pipe Name: DRIVER8156677015
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER9
Description: rte5_cpu0
Directory: c:\log\n1_rte50.log
Machine: n1_rte5
Parameter Set: PARAM9
Index: 800000000
Seed: 28814
Configured Users: 2700
Pipe Name: DRIVER9156714062
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Number of User groups: 42

APPENDIX C – TUNABLE PARAMETERS

Driver Engine: DRIVER1
IIS Server: n1_clienta
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 1 - 108
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER8
IIS Server: n1_clientd
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 2593 - 2700
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER8
IIS Server: n1_cliente
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 3349 - 3456
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER4
IIS Server: n1_clientd
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 2377 - 2484
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER8
IIS Server: n1_clientc
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 1891 - 1944
w_id Max Warehouse: 3780
Scale: Normal
User Count: 540
District id: 1
Scale Down: No

Driver Engine: DRIVER10
IIS Server: n1_clientd
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 2701 - 2808
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER9
IIS Server: n1_clienta
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 433 - 540
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER2
IIS Server: n1_clientd
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 2269 - 2376
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER9
IIS Server: n1_clientb
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 1189 - 1296
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER9
IIS Server: n1_clientc
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 1945 - 1998

APPENDIX C – TUNABLE PARAMETERS

w_id Max Warehouse: 3780
Scale: Normal
User Count: 540
District id: 1
Scale Down: No

Driver Engine: DRIVER10
IIS Server: n1_cliente
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 3457 - 3564
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER6
IIS Server: n1_clientd
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 2485 - 2592
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER10
IIS Server: n1_clientc
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 1999 - 2052
w_id Max Warehouse: 3780
Scale: Normal
User Count: 540
District id: 1
Scale Down: No

Driver Engine: DRIVER12
IIS Server: n1_cliente
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 3565 - 3672
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER11
IIS Server: n1_clientc
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 2053 - 2106
w_id Max Warehouse: 3780
Scale: Normal
User Count: 540
District id: 1
Scale Down: No

Driver Engine: DRIVER4
IIS Server: n1_clientc
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 1675 - 1728
w_id Max Warehouse: 3780
Scale: Normal
User Count: 540
District id: 1
Scale Down: No

Driver Engine: DRIVER11
IIS Server: n1_clienta
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 541 - 648
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER11
IIS Server: n1_clientb
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 1297 - 1404
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER13
IIS Server: n1_clientc
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 2161 - 2214
w_id Max Warehouse: 3780
Scale: Normal
User Count: 540
District id: 1
Scale Down: No

APPENDIX C – TUNABLE PARAMETERS

Driver Engine: DRIVER13
IIS Server: n1_clienta
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 649 - 756
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER1
IIS Server: n1_clientb
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 757 - 864
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER13
IIS Server: n1_clientb
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 1405 - 1512
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER6
IIS Server: n1_cliente
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 3241 - 3348
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER14
IIS Server: n1_clientc
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 2215 - 2268
w_id Max Warehouse: 3780
Scale: Normal
User Count: 540
District id: 1
Scale Down: No

Driver Engine: DRIVER14
IIS Server: n1_clientd
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 2917 - 3024
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER2
IIS Server: n1_cliente
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 3025 - 3132
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER2
IIS Server: n1_clientc
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 1567 - 1620
w_id Max Warehouse: 3780
Scale: Normal
User Count: 540
District id: 1
Scale Down: No

Driver Engine: DRIVER12
IIS Server: n1_clientc
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 2107 - 2160
w_id Max Warehouse: 3780
Scale: Normal
User Count: 540
District id: 1
Scale Down: No

Driver Engine: DRIVER3
IIS Server: n1_clientb
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 865 - 972

APPENDIX C – TUNABLE PARAMETERS

w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER3
IIS Server: n1_clienta
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 109 - 216
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER3
IIS Server: n1_clientc
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 1621 - 1674
w_id Max Warehouse: 3780
Scale: Normal
User Count: 540
District id: 1
Scale Down: No

Driver Engine: DRIVER4
IIS Server: n1_cliente
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 3133 - 3240
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER5
IIS Server: n1_clientb
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 973 - 1080
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER14
IIS Server: n1_cliente
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 3673 - 3780
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER5
IIS Server: n1_clienta
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 217 - 324
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER1
IIS Server: n1_clientc
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 1513 - 1566
w_id Max Warehouse: 3780
Scale: Normal
User Count: 540
District id: 1
Scale Down: No

Driver Engine: DRIVER5
IIS Server: n1_clientc
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 1729 - 1782
w_id Max Warehouse: 3780
Scale: Normal
User Count: 540
District id: 1
Scale Down: No

Driver Engine: DRIVER6
IIS Server: n1_clientc
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 1783 - 1836
w_id Max Warehouse: 3780
Scale: Normal
User Count: 540
District id: 1
Scale Down: No

APPENDIX C – TUNABLE PARAMETERS

Driver Engine: DRIVER7
IIS Server: n1_clientb
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 1081 - 1188
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER7
IIS Server: n1_clienta
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 325 - 432
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER12
IIS Server: n1_clientd
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 2809 - 2916
w_id Max Warehouse: 3780
Scale: Normal
User Count: 1080
District id: 1
Scale Down: No

Driver Engine: DRIVER7
IIS Server: n1_clientc
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 1837 - 1890
w_id Max Warehouse: 3780
Scale: Normal
User Count: 540
District id: 1
Scale Down: No

Number of Parameter Sets: 6

PARAM1
New Parameter Set

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	44.83	12.04	18.02	0.10	5.00	0.10
Payment	43.05	12.04	3.02	0.10	5.00	0.10
Delivery	4.05	5.04	2.02	0.10	5.00	0.10
Stock Level	4.04	5.04	2.02	0.10	20.00	0.10
Order Status	4.03	10.04	2.02	0.10	5.00	0.10

~Default
Default Parameter Set

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	44.80	12.05	18.01	0.10	5.00	0.10
Payment	43.05	12.05	3.01	0.10	5.00	0.10
Delivery	4.05	5.05	2.01	0.10	5.00	0.10
Stock Level	4.05	5.05	2.01	0.10	20.00	0.10
Order Status	4.05	10.05	2.01	0.10	5.00	0.10

PARAM2
New Parameter Set

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	44.83	12.05	18.01	0.10	5.00	0.10
Payment	43.05	12.05	3.01	0.10	5.00	0.10
Delivery	4.05	5.05	2.01	0.10	5.00	0.10
Stock Level	4.04	5.05	2.01	0.10	20.00	0.10
Order Status	4.03	10.05	2.01	0.10	5.00	0.10

PARAM3
New Parameter Set

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	44.84	12.05	18.01	0.10	5.00	0.10
Payment	43.04	12.05	3.01	0.10	5.00	0.10
Delivery	4.05	5.05	2.01	0.10	5.00	0.10
Stock Level	4.04	5.05	2.01	0.10	20.00	0.10
Order Status	4.03	10.05	2.01	0.10	5.00	0.10

PARAM4
New Parameter Set

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	44.85	12.05	18.01	0.10	5.00	0.10
Payment	43.04	12.05	3.01	0.10	5.00	0.10
Delivery	4.04	5.05	2.01	0.10	5.00	0.10
Stock Level	4.04	5.05	2.01	0.10	20.00	0.10
Order Status	4.03	10.05	2.01	0.10	5.00	0.10

PARAM9
New Parameter Set

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	44.84	12.05	18.01	0.10	5.00	0.10
Payment	43.04	12.05	3.01	0.10	5.00	0.10
Delivery	4.05	5.04	2.01	0.10	5.00	0.10
Stock Level	4.04	5.04	2.01	0.10	20.00	0.10
Order Status	4.03	10.06	2.01	0.10	5.00	0.10

Please note that PARAM9 is the one that was used for the performance run.

Appendix D Disk Storage

TPC-C 180 Day Space Requirements						
Warehouses	**3430**				**TpmC**	**43,046.55**
Table	**Rows**	**Data KB**	**Index KB**	**Extra 5% KB**	**8hr Space**	**Total Space KB**
Warehouse	3430	368	48	21		437
District	34300	3,816	56	194		4066
Customer	102900000	74,836,368	4,462,536	3,964,945		83263849
History	102900000	5,716,680	96		1,183,311	5716776
NewOrder	30870000	488,064	1,176			489240
Orders	102900000	3,154,024	1,434,312		1,678,959	4588336
OrderLine	1028999831	64,312,496	136,176		12,919,546	64448672
Item	100000	9,528	72	480		10080
Stock	343000000	109,760,000	205,176	5,498,259		115463435
Total		258,281,344	6,239,648	9,463,898	15,781,816	273,984,890
	MB					
Dynamic Space	71,468	Sum of Data for Order, Orderline and History				
Static Space	196,095	Sum of Data+Index+5%-Dynamic Space				
Free Space	na	Total Allocated Spac - (Dynamic + Static Space)				
Daily Growth	14,351	(Dynamic Space/(W*62.5))*tpmc				
Daily Spread	-	(Free Space -1.5*Daily Growth) Zero Assumed				
180 Day Space MB	2,779,237					
180 Day Space GB	**2,714.10**	**GB**				
Log Size	48,000	MB				
KB Per New Order	5.1213	KB			**Disk Size**	**Formatted Size**
8 hr log MB	103,338	MB			18GB	16.900
8 hr log GB	**100.9161**	**GB**			9GB	8.470
					4GB	3.999
Space Usage	**GB Needed**	**Disks Measured**	**GB Measured**	**Disks Priced**	**GB Priced**	
180 Day Space DB	2,714.10	36	608.40	36	608.40	
		240	2032.80	240	2032.80	
		0	0.00	0	0.00	
Total DB		**276.00**	**2641.20**		**2641.20**	
8-hr log + mirror	201.8321	16	270.40	16	270.40	
OS, Swap	-	1	8.46	1	8.46	
			included w/log			
Total Storage	**2,915.93**	**GB**	**2,920.06**	**GB**	**2,920.06**	**GB**

tpmC	43,046.55									
	Data Before KB	**Index Before KB**	**Data After KB**	**Index After KB**	**Data Grow KB**	**Index Grow KB**	**Total Grow KB**	**KB/New-Order**	**8-Hr Growth KB**	**8-Hr Growth MB**
History	5,834,424	160	6,009,640	160	175,216	-	175,216	0.0573	1,183,310.66	1,155.58
Order	3,428,416	2,872,576	3,675,560	2,874,040	247,144	1,464	248,608	0.0813	1,678,959.09	1,639.61
Order-Line	66,077,232	272,320	67,990,264	272,320	1,913,032	-	1,913,032	0.6253	12,919,545.87	12,616.74
										15,411.93
	sum(*) Before		**sum(*) After**		**Num New-Order**					
d_next_o_id	104,999,406		108,058,935		3,059,529					
						from batch report				
	Before MB		**After MB**		**Grow MB**			**KB/New-Order**	**8-Hr Growth MB**	**8-Hr Growth GB**
Log	13402.59		28704.13		15301.54			5.1213	103,338.04	100.92
								5,244.2157		
119999.9922	11.168825		23.920111							
Database tpcc log used (%)		12.751286								

Appendix E Price Quotations

Software House International

Pricing Proposal

SHI Account Exec: Matthew Martin Telephone: (800) 766-6357 ext. 106 Fax: (408) 232-2585

HP Lxr 8500 Quote - July 10, 2001				
Description	**Part #**	**Qty**	**Price**	**Extended**
HP NetServer LXr 8500 Intel Pentium III Xeon 700MHz	P1765AV	1	$17,550	$17,550
Intel Pentium III Xeon 700MHz 2Mbyte L2	P1763AV	7	$3,829	$26,803
512MB Dimm for LXr 8500 (to 8GB)	D7138AV	16	$1,364	$21,824
HP NetRAID4M RAID controller w/64MB cache	D9161A	7	$1,647	$11,529
HP 9GB, 10krpm Hot-swap disk module	P1168A	1	$262	$262
HP NetServer 10/100TX PCI LAN Adapter	D5013B	1	$69	$69
HP 15" Color Monitor	D2828A	1	$185	$185
HP NetServer mini-DIN keyboard and mouse	D4950B/C3751B	1	$79	$79
HP Rack System/E41 (41 Units usable space)	J1500A	2	$1,590	$3,180
HP Power Distribution Unit 120-240V	E5929A	4	$234	$936
HP SureStore DAT24i Internal Tape Drive	C1555D	1	$777	$777
APC Smart-UPS 3000 + Spares	588293	3	$1,340	$4,020
HP LXr 8500 Support Pack 3 years, 24x7, 4 hr response	H5515A	1	$2,193	$2,193
HP NetServer Rack Storage/12	D5989B	25	$1,890	$47,250
HP RS/12 Support Pack 3 years, 24x7, 4 hr response	H5513A	25	$1,450	$36,250
HP 9GB, 10krpm Hot-swap disk module	P1168A	240	$262	$62,880
HP 18GB, 10krpm Hot-swap disk module	P1166A	52	$530	$27,560
HP SCSI Cable 2.5m UDHTS 68/HDTS 68	D6020A	25	$84	$2,100
HP Netserver E60 Pentium III 600MHz	D9128A	5	$1,270	$6,350
HP E60 Support Pack 3 years, 24x7, 4 hr response	H5512A	5	$988	$4,940
128MB DRAMs for E60	D7156A	20	$205	$4,100
HP NetServer 10/100TX PCI LAN Adapter	D5013B	5	$69	$345
HP 15" Color Monitor	D2828A	5	$185	$925
HP Procurve Switch 2424M, Lifetime Warranty + 10% spares	J4093A	8	$940	$7,520
Microsoft Windows 2000 Advanced Server, 25 Licences		1	$2,399	$2,399
Microsoft SQL Server 2000 Enterprise Edition per processor license unlimited users (open program level C)	810-00846	8	$16,541	$132,328
Microsoft Windows 2000 Server	C11-0016	5	$790	$3,950
Microsoft Visual C++ Professional 6.0	716856	1	$449	$449
		Total		**$428,753**
Quote Good for Ninety Days				