TPC Benchmark® C
Full Disclosure Report

HP Netserver LH 6000
using Microsoft SQLServer 2000 Enterprise Edition
on Microsoft Windows 2000 Advanced Server

First Edition
August 23rd, 2001

First Edition - August 23rd, 2001
First Printing.

Hewlett-Packard Company believes that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. Hewlett-Packard Company assumes no responsibility for any errors that may appear in this document.

The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, Hewlett-Packard Company provides no warranty of the pricing information in this document.

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

All performance data contained in this report was obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly. Hewlett-Packard Company does not warrant or represent that a user can or will achieve similar performance expressed in transactions per minute (tpmC ®) or normalized price/performance ($/tpmC ®). No warranty of system performance or price/performance is expressed or implied in this report.

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 LH 6000 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 LH 6000 system. The Standard System Summary is given below.

Company Name	System Name	Database Software	Operating System
Hewlett-Packard Company	HP Netserver LH 6000	Microsoft SQLServer 2000 Enterprise Edition	Microsoft Windows 2000 Advanced Server
Total System Cost	**TPC-C Throughput**	**Price/Performance**	**Availability Date**
$333,572.00	37596.34 tpmC	$8.87 per tpmC	August 23rd, 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 Lorna Livingtree of Performance Metrics, Inc. to verify compliance with the relevant TPC specifications.

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

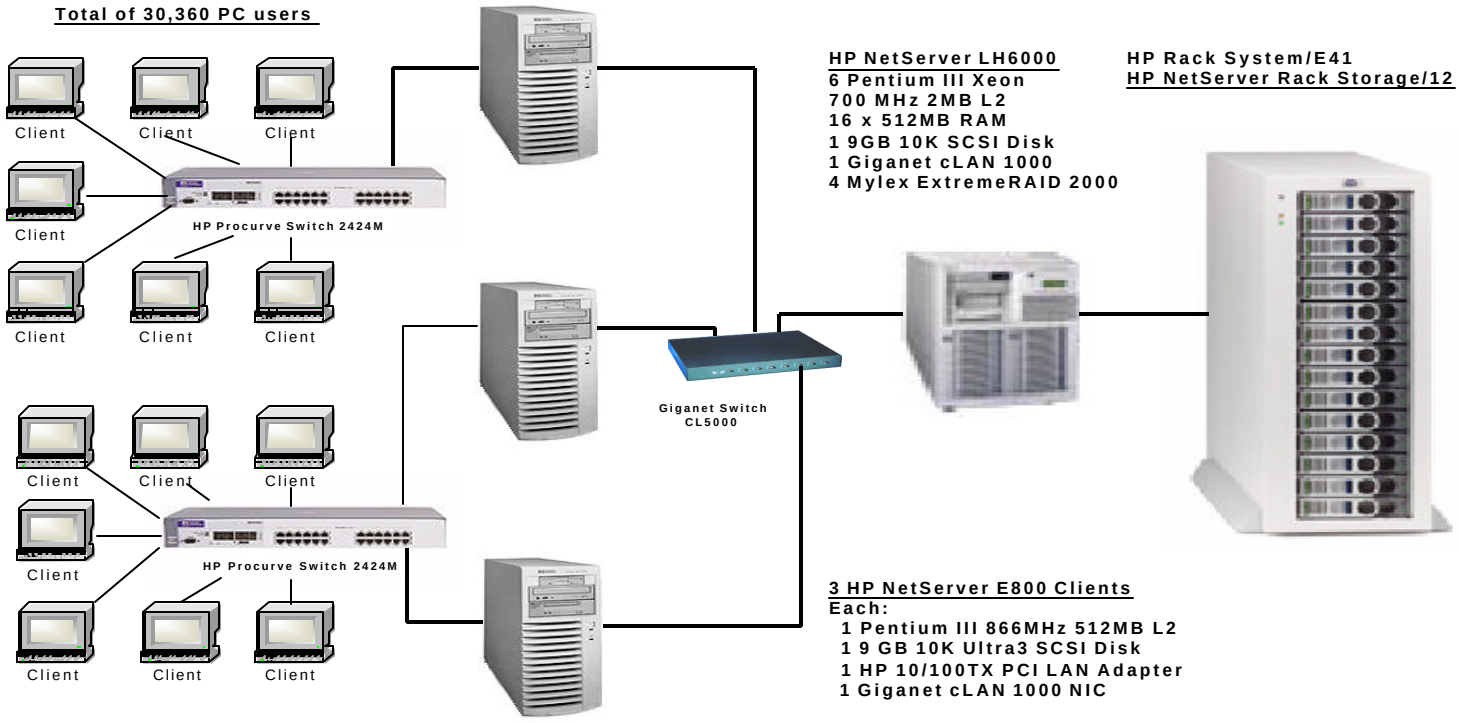
HP Netserver LH 6000 Client/Server

TPC-C Rev 5.0

Report Date: August 23rd, 2001

Total System Cost	TPC Throughput	Price/Performance	Availability Date
$333,572	37596.34 tpmC	$8.87 per tpmC	August 23rd, 2001

Processors	Database Manager	Operating System	Other Software	Number of Users
6 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	30360



System Components	Server		Each Client	
	Qty	Type	Qty	Type
Processors	6	700MHz Intel Pentium III Xeon	1	866MHz Pentium III
Cache Memory	each	2 Mbyte L2 cache		512kbyte L2 Cache
Memory	16	512 Mbyte		512 Mbyte
Disk Controllers	4	Mylex eXtreme2000	1	HP SCSI-2 Controller
Disk Drives	156	HP hot-swap 18GB SCSI	1	9 Gbyte disk
Total Storage		2644.2 Gbyte		8.46 Gbyte
Tape Drives	0	HP Surestore DAT24i		
Terminals	1	Console Terminal	1	Console Terminal

HP Netserver LH 6000
6P/700-2MB/8GB

TPC-C Rev 5.0 Executive Summary
Page 2 of 2

Report Date: 23-Aug-01

Description	Part Number	Brand	Price Key	Unit Price	Qty	Extended Price	Maint. Price
LH 6000r with six Pentium III Xeon 700MHz 2MB L2 cache	P4583AV		1	18,960	1	18,960	
HP LH 6000 Support Pack, 3 years, 24x7, 4 hour response	H3127A			1,715	1		1,715
512 MB DIMM kit	D8267AV		1	610	16	9,760	
HP 15" Color Monitor	D8896A		1	144	1	144	
HP NetServer mini-DIN keyboard and mouse	D4950B/C3751B		1	79	1	79	
HP 9GB 10K HotSwap Wide Ultra2 SCSI Disk	P1168A		1	262	1	262	
HP Rack System/E41 (41 Units usable space)	J1500A		1	1,590	1	1,590	
HP Power Distribution Unit 120-240V	E5929A		1	234	2	468	
HP SureStore DAT24i Internal Tape Drive	C1555D		1	759	1	759	
Mylex eXtremeRAID 2000 (incl. spares)	E2000-4-32ME	Mylex	1	1,872	6	11,232	
Giganet cLAN-1000 Host Adaptor	cLAN-1000	Emulex	2	795	1	795	
Giganet cLAN 2M Cable	cLAN-A0211	Emulex	2	95	1	95	
Giganet cLAN-5000 8-port Switch (incl. spares)	cLAN-5000	Emulex	2	6,250	3	18,750	
APC Smart-UPS 3000 (incl. spares)	588293	APC	1	1,340	3	4,020	
					Server Hardware Subtotal	**66,914**	**1,715**
HP Netserver Rack Storage/12	D5989B		1	1,890	13	24,570	
HP RS/12 Support Pack, 3 years, 24x7, 4 hour response	H5513A			1,450	13		18,850
HP SCSI Cable 2.5m UDHTS 68/HDTS 68	D6020A		1	84	13	1,092	
HP 18GB, 15krpm Hot-swap disk module	D9421A		1	669	156	104,364	
					Storage Subtotal	**130,026**	**18,850**
Microsoft SQL Server 2000 Enterprise Edition (per processor)	810-00846	MS	1	14,869	6	89,214	6,285
Microsoft Windows 2000 Advanced Server	C10-00475	MS	1	2,160	1	2,160	
					Server Software Subtotal	**91,374**	**6,285**
HP Netserver E800 Pentium III 866MHz	P2457A		1	832	3	2,496	
HP E800 Support Pack, 3 years, 24x7, 4 hour response	H4636E			745	3		2,235
128MB DRAMs for E800	D8265A		1	157	12	1,884	
HP 9GB 10K Ultra3 SCSI Disk	P1214A		1	167	3	501	
HP 15" Color Monitor	D8896A		1	144	3	432	
Giganet cLAN-1000 Host Adaptor (incl. spares)	cLAN-1000	Emulex	2	795	5	3,975	
Giganet cLAN 2M Cable (incl. spares)	cLAN-A0211	Emulex	2	95	5	475	
					Client Hardware Subtotal	**9,763**	**2,235**
Microsoft Windows 2000 Server	C11-00016	MS	1	839	3	2,517	
Microsoft Visual C++	716856	MS	1	449	1	449	
					Client Software Subtotal	**2,966**	**0**
HP Procurve Switch 2424M (incl. spares)	J4093A		1	861	4	3,444	
					Connectivity Subtotal	**3,444**	**0**
					Total	**$304,487**	**$29,085**

Notes: Price key: 1 = Software House, 2 = Emulex

Audited by Performance Metrics Inc.

5-yr Cost of Ownership: **$333,572**
tpmC: **37,596.34**
$/tpmC: **$8.87**

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 LH 6000

MQTH, Computed Maximum Qualified Throughput **37,596.34 tpmC**

Response Times (in seconds)

	90th %-ile	Maximum	Average
New-Order	1.37s	15.20s	0.64s
Payment	1.24s	14.52s	0.52s
Order-Status	1.28s	11.13s	0.56s
Delivery (interactive portion)	0.11s	3.75s	0.11s
Delivery (deferred portion)	0.63s	5.99s	0.37s
Stock-Level	2.75s	12.37s	1.66s
Menu	0.11s	3.91s	0.11s

Response time delay added for emulated components 0.1 seconds

Transaction Mix, in percent of total transactions

New-Order	44.91%
Payment	43.03%
Delivery	4.01%
Stock-Level	4.02%
Order-Status	4.03%

Keying/Think Times

	Keying Time			Think Time		
	Min	Avg	Max	Min	Avg	Max
New-Order	18s	18.01s	18.03s	0s	12.06s	120.5s
Payment	3s	3.02s	3.03s	0s	12.04s	120.5s
Order-Status	2s	2.02s	2.03s	0s	10.05s	100.5s
Delivery (interactive)	2s	2.02s	2.03s	0s	5.05s	50.5s
Stock-Level	2s	2.02s	2.03s	0s	5.05s	50.5s

Test Duration

Ramp up time	72 minutes
Measurement interval	120 minutes
Transactions during measurement interval	10,449,438
Ramp down time	3 minutes

Checkpointing

Number of checkpoints in measurement interval	4
Checkpoint Interval	30 minutes

Table of Contents

Preface

Document Structure

This is the full disclosure report for a benchmark test of the HP Netserver LH 6000 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 LH 6000. 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 LH 6000 was designed for the Intel Pentium III Xeon processor. The HP Netserver LH 6000 used in this test was powered by six 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 four Mylex eXtremeRAID 2000 Disk Array Controllers (DACs), which were plugged into four PCI slots on the motherboard. The database log drives consisted of six pairs of 18GB 15krpm Ultra Wide hard disks attached to one DAC. The TPC-C database storage consisted of 144 HP 18GB 15krpm hard drives. The 18GB drives on each DAC were equally distributed across the 4 SCSI channels. The channels were striped using the Mylex Global Array Manager and RAID EzAssist configuration utilities. Controller write-back caching and read ahead were specifically disabled for the PCI DACs.

Each of the three clients is a HP Netserver E 800 with a single Pentium III 866MHz, 512 MB RAM, one 9GB SCSI hard disk, running Microsoft Windows 2000 Server.

The server, clients and the simulated users were networked together via two HP Procurve 2424M 10/100baseT switches. Eleven remote terminal emulators (RTEs) emulated 30,360 users executing the standard TPC-C workload. The switches are connected to each client machine through 100base T links and to all eleven RTEs at 10Mbit/sec. Each client had two LAN adapters, an embedded 10/100BaseT and a cLAN interface. On each client, the cLAN adapter was connected to the server through a Giganet link and the embeded adapter was connected via ProCurve switch to eleven RTEs at 10Mbit/sec, half-duplex. Three clients driven through eleven network segments each, provided 33 network segments and 920 emulated users per network segment for the 30,360 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 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 60 day space calculations.

Configuration Diagrams

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

The measured and priced client/server configurations are shown in Figures 1 and 2.

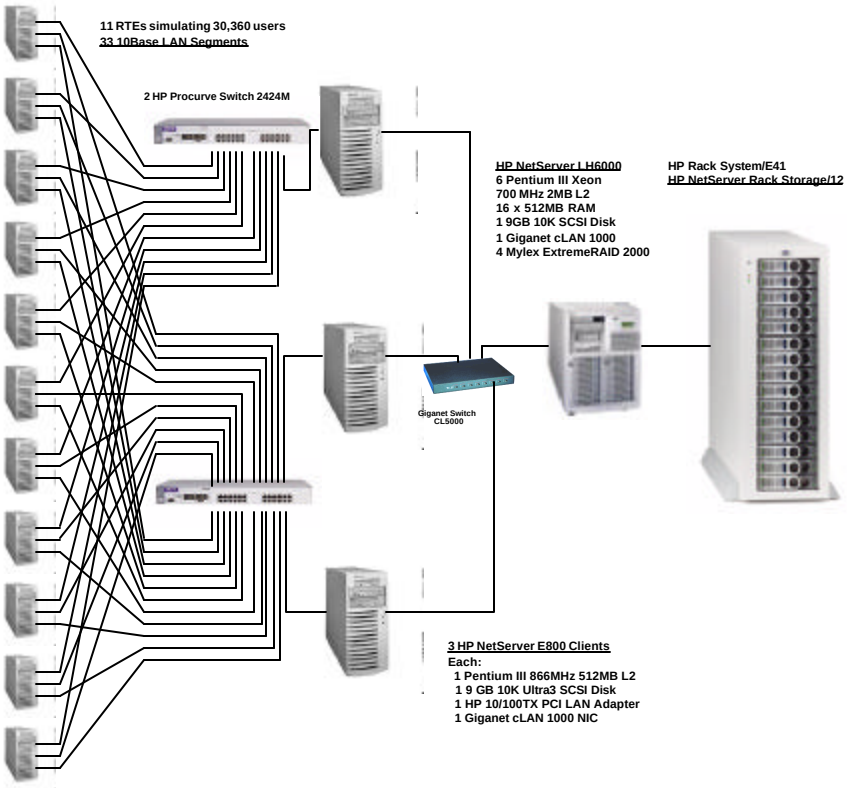


Figure 1. Measured Configuration

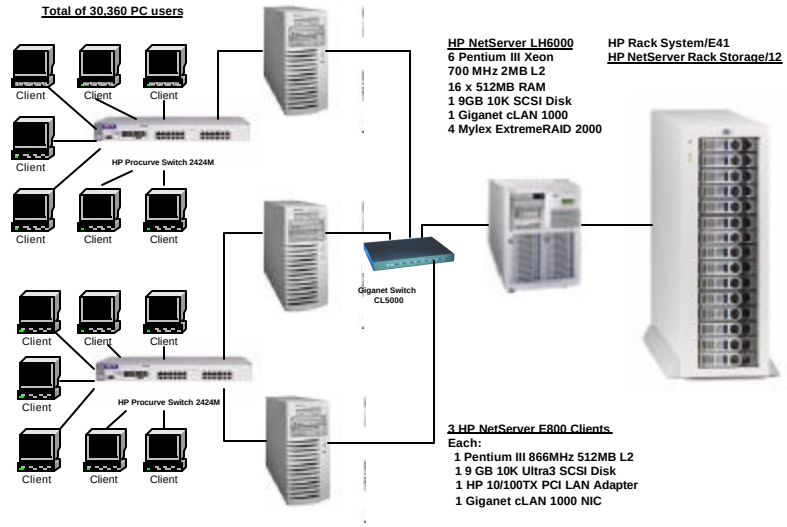


Figure 2. Priced Configuration

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 257 disks, which included 144 18GB Hot Swap disk drives for data, 12 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, and audited independently.

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	99.01%
	Remote warehouse items	0.99%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse	85.00%
	Remote warehouse	15.00%
	Non primary key access	60.03%
Order Status	Non primary key access	59.91%
Delivery	Skipped transactions	0
Transaction Mix	New Order	44.91%
	Payment	43.03%
	Delivery	4.01%
	Stock Level	4.02%
	Order Status	4.03%

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 LH 6000. A fully scaled database was used except for the durability tests of durable media failure. The test was performed on a database scaled to 500 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 500 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 500 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 500 warehouse database for convenience. The standard driving mechanism was used to generate the transaction load of 5,000 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 500 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. 5,000 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 3036 warehouses in a single test. The standard driving mechanism was used to generate the transaction load of 30,360 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. 30,360 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 4 Mylex eXtreme2000 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 256 disks – 144 18GB for data, 12 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, and 3 were configured with 12 disk drives per RAID stripe and 4 spans for a total of 48 disk drives per data controller. DAC 4 was configured with 2 mirrored drives per RAID stripe and a 6 drive span for a total of 12 log drives. DAC 1, 2, and 3 contained 3 partitions: partition 1 for customer/stock, partition 2 for miscellaneous, and partition 3 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

<table>
<tr><td colspan="5">MYLEX CONFIGURATION</td><td>WIN2K DISK ADMINISTRATION</td></tr>
<tr><td colspan="5">Controller #1</td><td>Disk 1: 101.62 GB</td></tr>
<tr><td rowspan="2">SCSI ID</td><td colspan="4">Channels</td><td>Partition (RAID 1+0)</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>1</td></tr>
<tr><td>0</td><td>A0-1</td><td></td><td></td><td></td><td rowspan="14">E:
LOG
Unknown
101.62 GB</td></tr>
<tr><td>1</td><td>A0-2</td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>A1-1</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>A1-2</td><td></td><td></td><td></td></tr>
<tr><td>8</td><td>A2-1</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td>A2-2</td><td></td><td></td><td></td></tr>
<tr><td>10</td><td>A3-1</td><td></td><td></td><td></td></tr>
<tr><td>11</td><td>A3-2</td><td></td><td></td><td></td></tr>
<tr><td>12</td><td>A4-1</td><td></td><td></td><td></td></tr>
<tr><td>13</td><td>A4-2</td><td></td><td></td><td></td></tr>
<tr><td>14</td><td>A5-1</td><td></td><td></td><td></td></tr>
<tr><td>15</td><td>A5-2</td><td></td><td></td><td></td></tr>
</table>

MYLEX CONFIGURATION					WIN2K DISK ADMINISTRATION		
Controller #2					Disk 2: 813.00 GB		
SCSI ID	Channels				Partitions (RAID 0)		
	0	1	2	3	1	2	3
0	A0-1	A0-13	A1-9	A2-5	F: MISC1 Unknown 30.76 GB	M: CS1 Unknown 69.33 GB	V: Backup NTFS 712.90 GB
1	A0-2	A0-14	A1-10	A2-6			
2	A0-3	A0-15	A1-11	A2-7			
3	A0-4	A0-16	A1-12	A2-8			
8	A0-5	A1-1	A1-13	A2-9			
9	A0-6	A1-2	A1-14	A2-10			
10	A0-7	A1-3	A1-15	A2-11			
11	A0-8	A1-4	A1-16	A2-12			
12	A0-9	A1-5	A2-1	A2-13			
13	A0-10	A1-6	A2-2	A2-14			
14	A0-11	A1-7	A2-3	A2-15			
15	A0-12	A1-8	A2-4	A2-16			

MYLEX CONFIGURATION					WIN2K DISK ADMINISTRATION		
Controller #3					Disk 3: 813.00 GB		
SCSI ID	Channels				Partitions (RAID 0)		
	0	1	2	3	1	2	3
0	A0-1	A0-13	A1-9	A2-5	G: MISC1 Unknown 30.76 GB	N: CS1 Unknown 69.33 GB	Freespace 712.90 GB
1	A0-2	A0-14	A1-10	A2-6			
2	A0-3	A0-15	A1-11	A2-7			
3	A0-4	A0-16	A1-12	A2-8			
8	A0-5	A1-1	A1-13	A2-9			
9	A0-6	A1-2	A1-14	A2-10			
10	A0-7	A1-3	A1-15	A2-11			
11	A0-8	A1-4	A1-16	A2-12			
12	A0-9	A1-5	A2-1	A2-13			
13	A0-10	A1-6	A2-2	A2-14			
14	A0-11	A1-7	A2-3	A2-15			
15	A0-12	A1-8	A2-4	A2-16			

MYLEX CONFIGURATION					WIN2K DISK ADMINISTRATION		
Controller #4					Disk 4: 813.00 GB		
SCSI ID	Channels				Partitions (RAID 0)		
	0	1	2	3	1	2	3
0	A0-1	A0-13	A1-9	A2-5	H: MISC1 Unknown 30.76 GB	O: CS1 Unknown 69.33 GB	Freespace 712.90 GB
1	A0-2	A0-14	A1-10	A2-6			
2	A0-3	A0-15	A1-11	A2-7			
3	A0-4	A0-16	A1-12	A2-8			
8	A0-5	A1-1	A1-13	A2-9			
9	A0-6	A1-2	A1-14	A2-10			
10	A0-7	A1-3	A1-15	A2-11			
11	A0-8	A1-4	A1-16	A2-12			
12	A0-9	A1-5	A2-1	A2-13			
13	A0-10	A1-6	A2-2	A2-14			
14	A0-11	A1-7	A2-3	A2-15			
15	A0-12	A1-8	A2-4	A2-16			

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	3,036
District	30,360
Customer	91,080,000
History	91,080,000
Orders	91,080,000
New Orders	27,324,000
Order Line	468,001,833
Stock	100,000
Item	303,600,000

No rows were deleted for the benchmark runs.

4.3 60 Day Space

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

4.3.1 Transaction Log Space Requirements

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

1. The free space on the logfile was queried using **dbcc sqlperf(logspace)**.
2. Transactions were run against the database with a full load of users.
3. The free space was again queried using **dbcc sqlperf(logspace)**.
4. The space used was calculated as the difference between the first and second query.
5. The number of NEW-ORDERS was verified from an RTE report covering the entire run.
6. The space used was divided by the number of NEW-ORDERS giving a space used per NEW-ORDER transaction.
7. The space used per transaction was multiplied by the measured tpmC rate times 480 minutes.

The result of the above steps yielded a requirement of 87.09 GB (including mirror) to sustain the log for 8 hours. Space available for the transaction log was 101.62 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 60 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 MSSQL_cs_fg and MSSQL_misc_fg. MSSQL_cs_fg consist of 3 partitions at 60 GB each and MSSQL_misc_fg consist of 3 partitions at 29 GB each as shown in the createdb.sql. The log was configured with 101 GB.

Chapter 5 Performance Metrics and Response Time

5.1 Throughput

Measured tpmC® must be reported.

Measured TpmC®: 37,596.34
Price per TpmC®: 8.87

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	0.64s	1.37s	15.20s
Payment	0.52s	1.24s	14.52s
Order-Status	0.56s	1.28s	11.13s
Delivery (interactive portion)	0.11s	0.11s	3.75s
Delivery (deferred portion)	0.37s	0.63s	5.99s
Stock-Level	1.66s	2.75s	12.37s
Menu	0.11s	0.11s	3.91s

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	3.02s	3.03s
Order Status	2s	2.02s	2.03s
Interactive Delivery	2s	2.02s	2.03s
Stock Level	2s	2.02s	2.03s

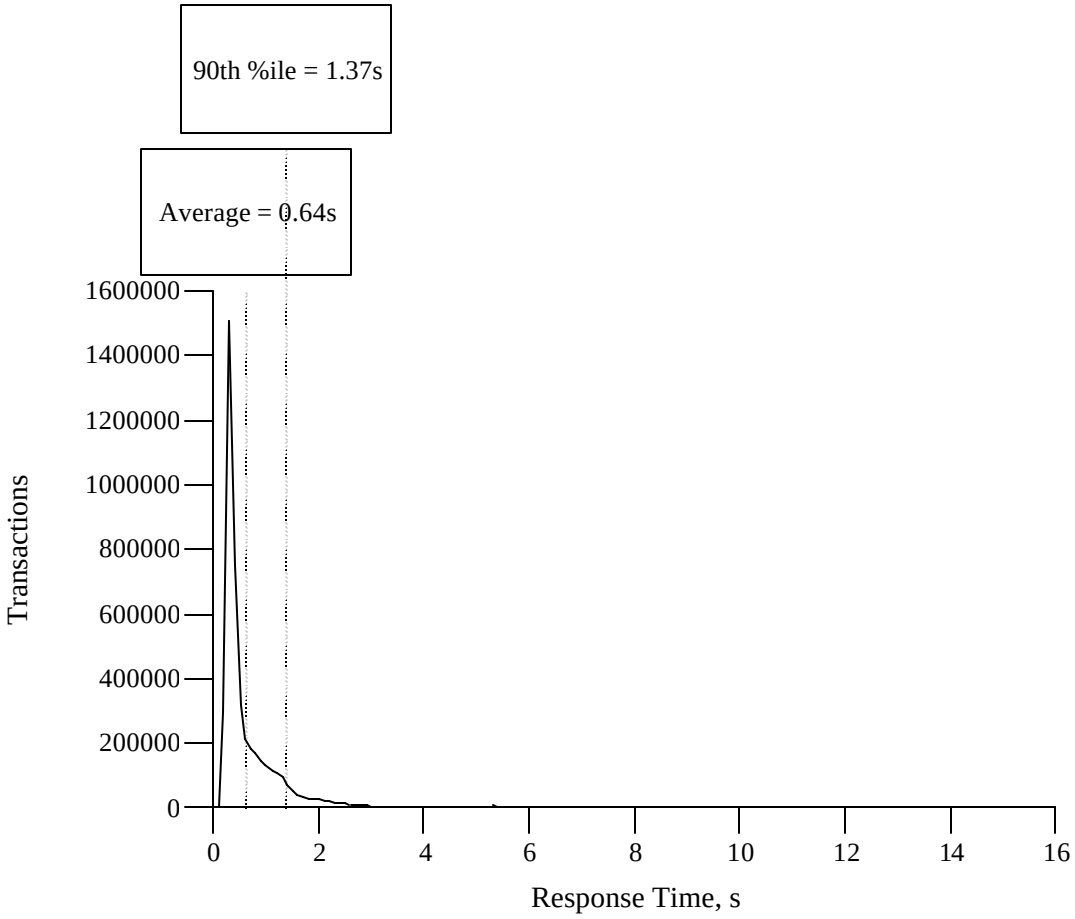
Table 6: Transaction Think Times

Think Times	Minimum	Average	Maximum
New Order	0s	12.06s	120.5s
Payment	0s	12.04s	120.5s
Order Status	0s	10.05s	100.5s
Interactive Delivery	0s	5.05s	50.5s
Stock Level	0s	5.05s	50.5s

5.4 Response Time Frequency

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

5.4.1 New Order Response Time



Response time frequency distribution for New Order transaction

Figure 3: New Order Response Time Distribution

5.4.2 Payment Response Time Distribution

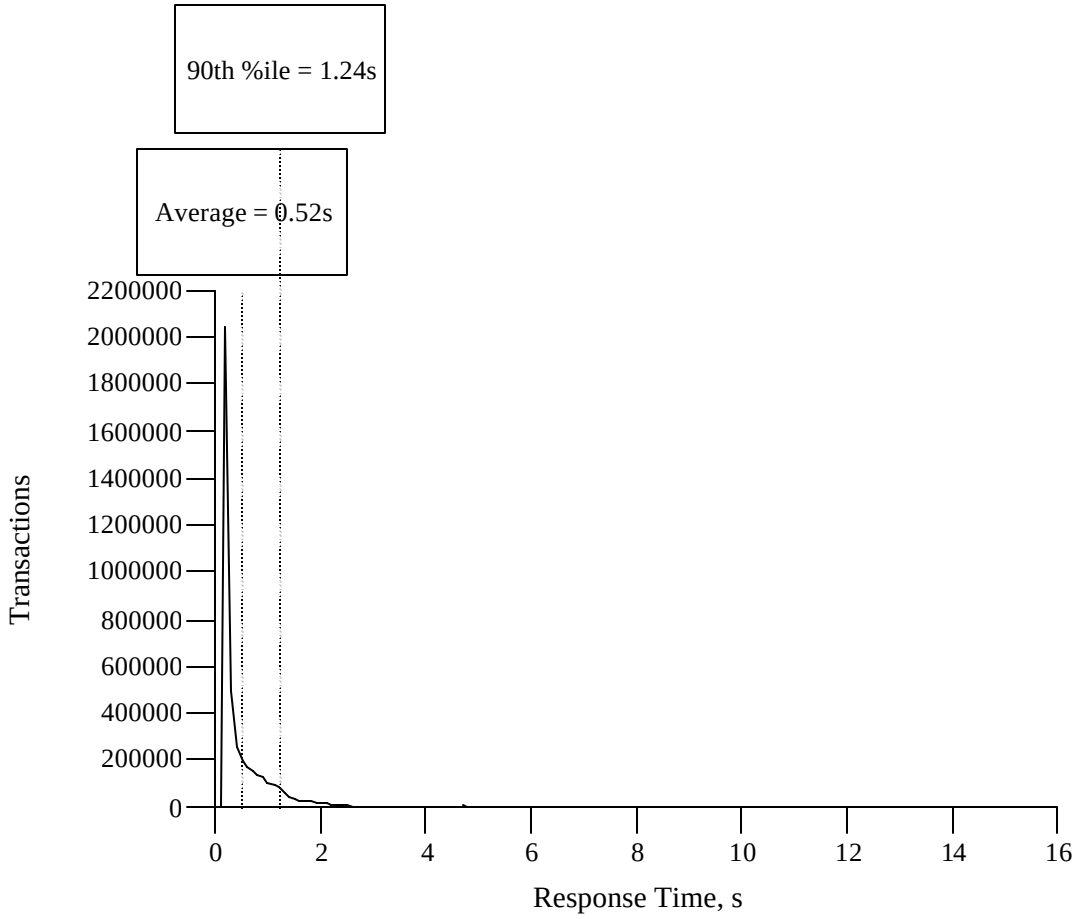
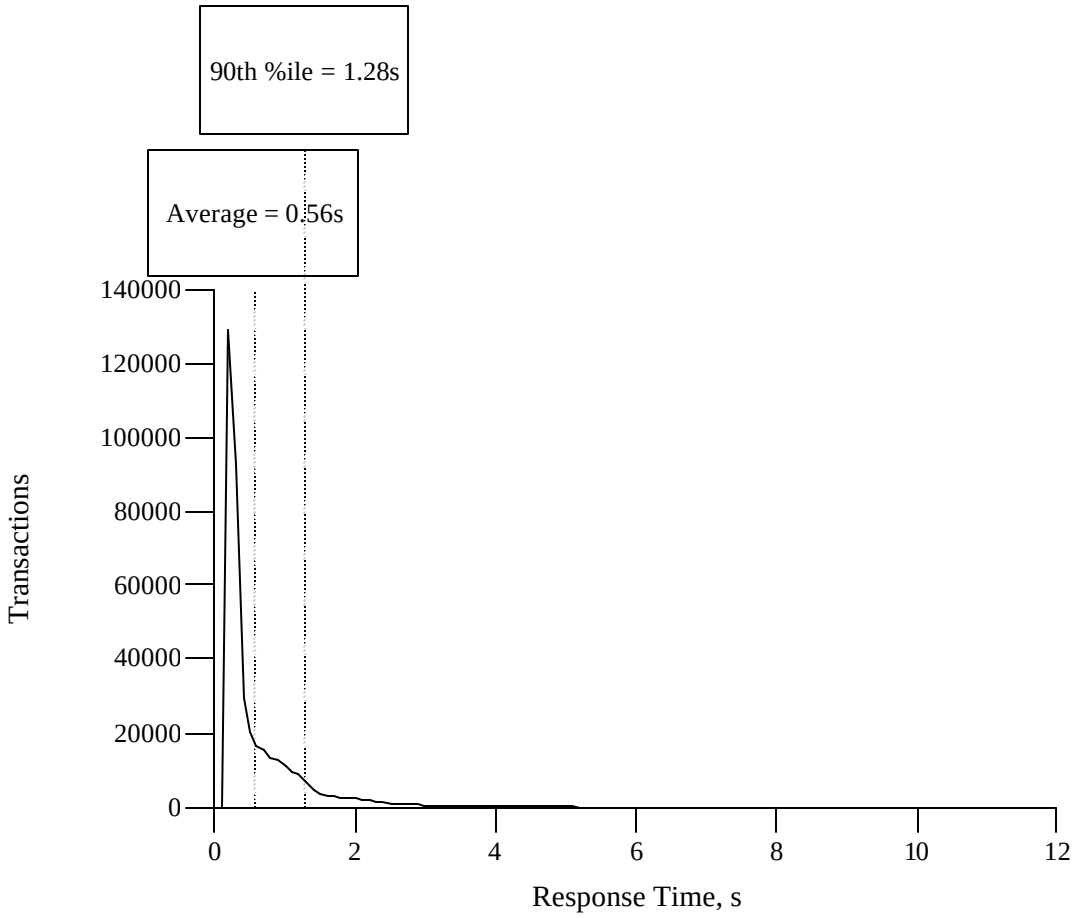


Figure 4: Payment Response Time Distribution

5.4.3 Order Status Response Time



Response time frequency distribution for Order Status transaction

Figure 5: Order Status Response Time Distribution

5.4.4 Delivery Response Time Distribution

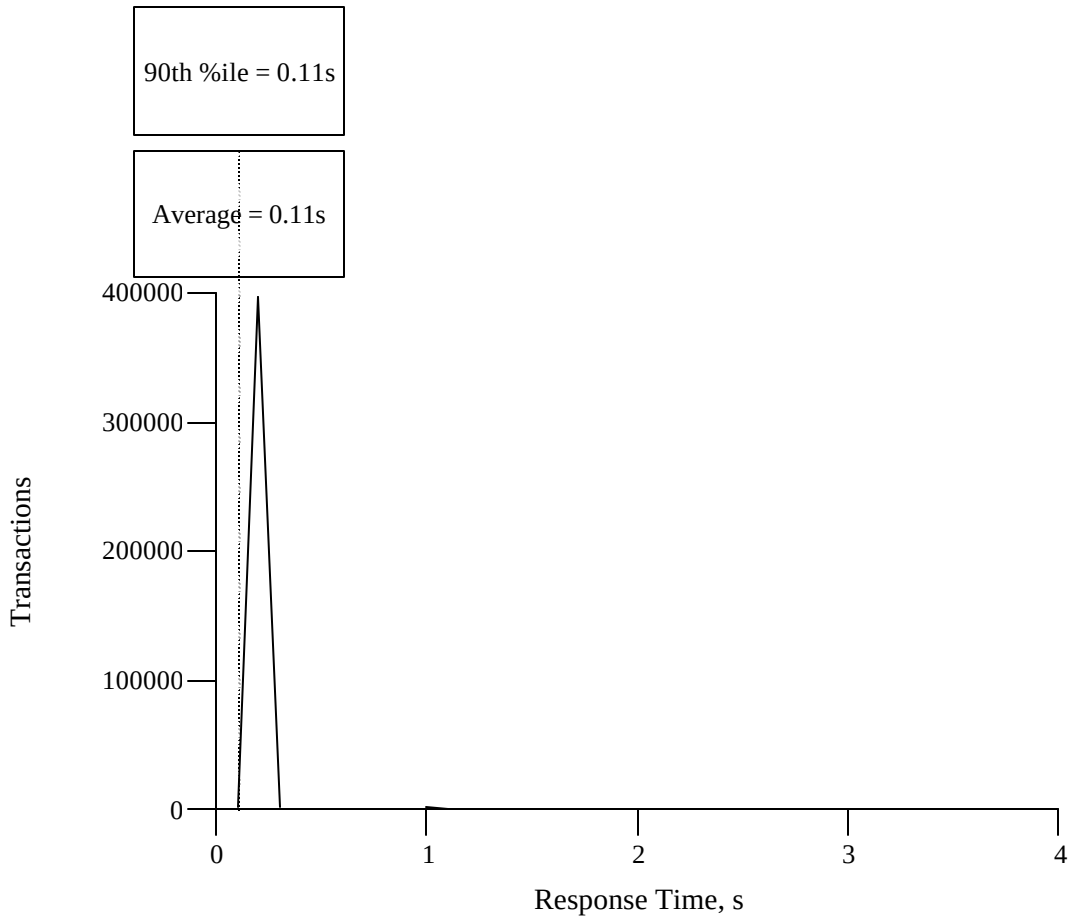
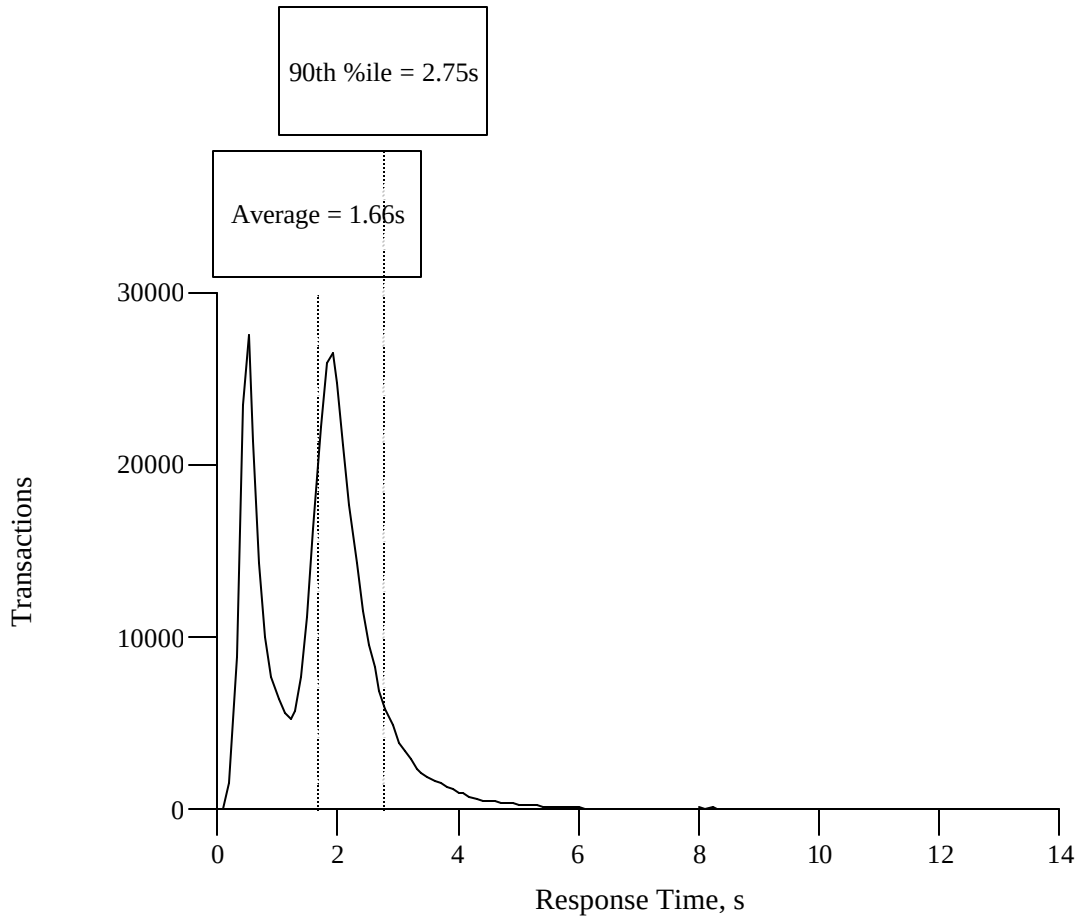


Figure 6: Delivery Response Time Distribution

5.4.5 Stock Level Response Time



Response time frequency distribution for Stock Level transaction

Figure 7: Stock Level Response Time Distribution

5.4.6 Response Time Versus Throughput

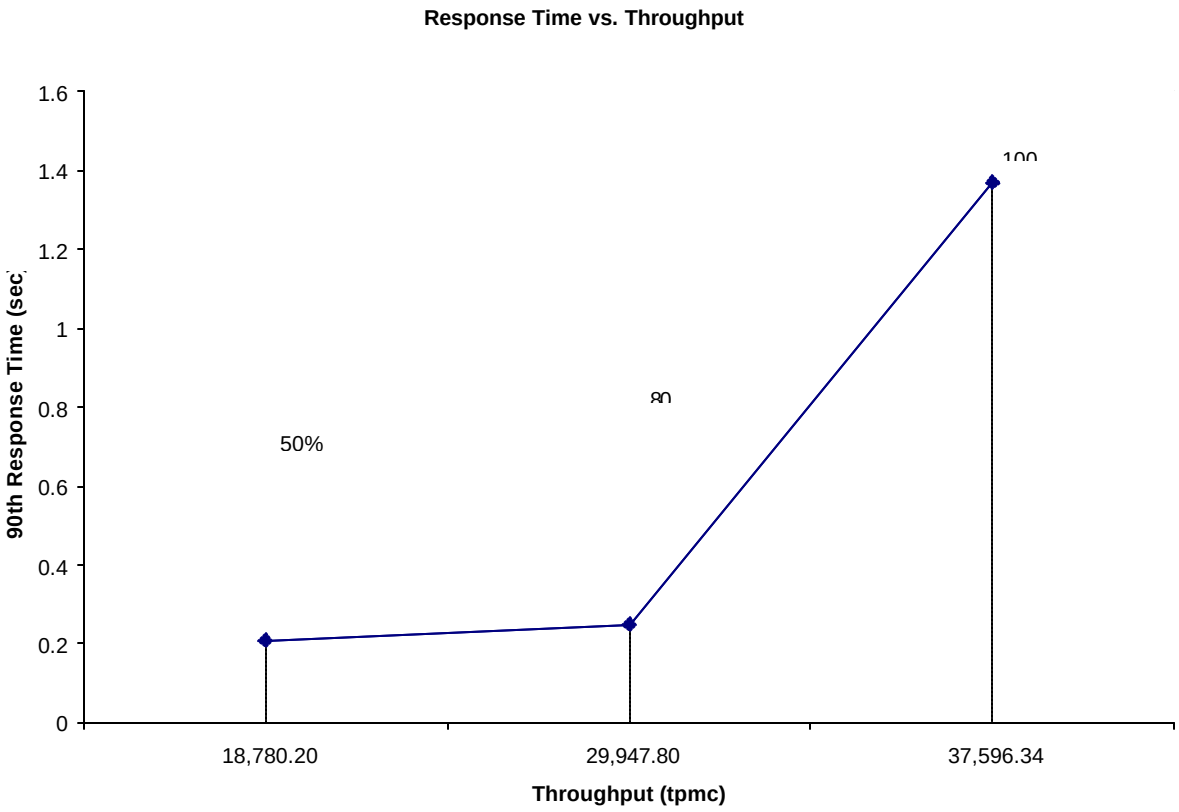
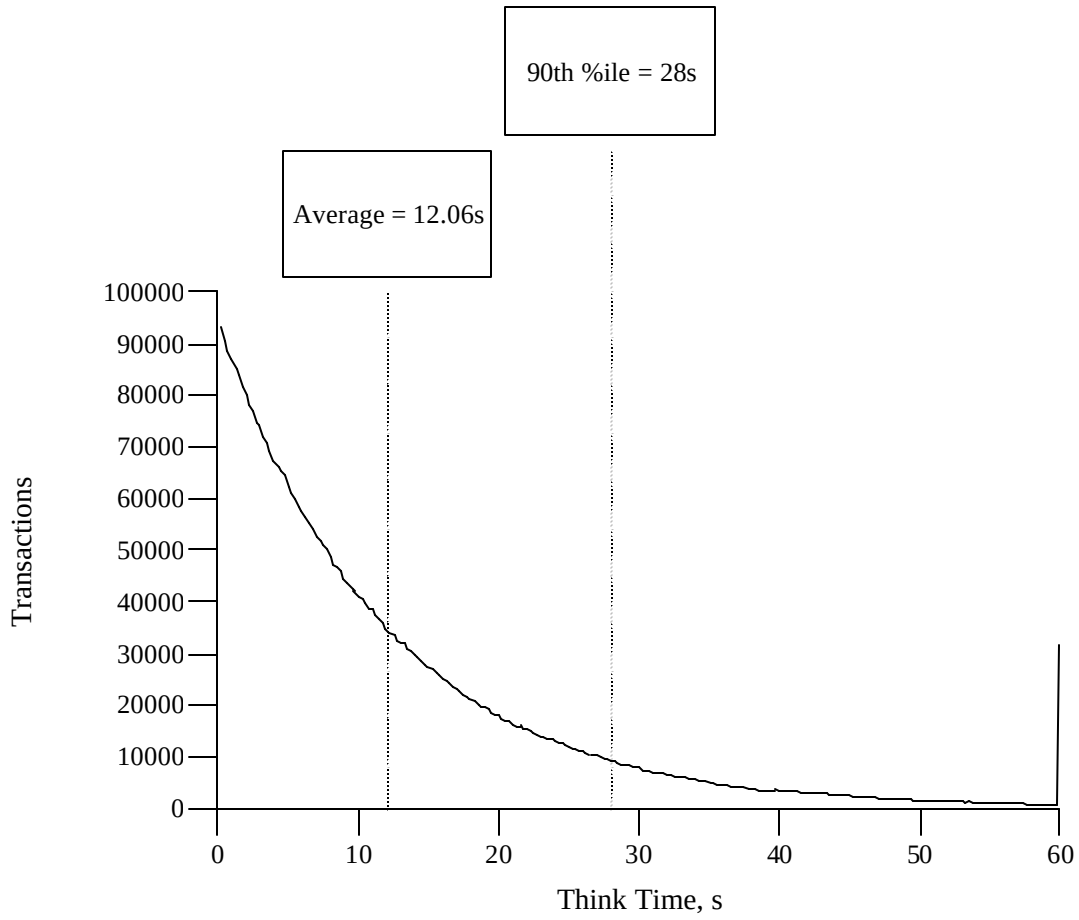


Figure 8: New Order Response Time Distribution

5.4.7 New Order Think Time Distribution



Think time frequency distribution for New Order transaction

Figure 9: New Order Think Time Distribution

5.4.8 Throughput Versus Time Distribution

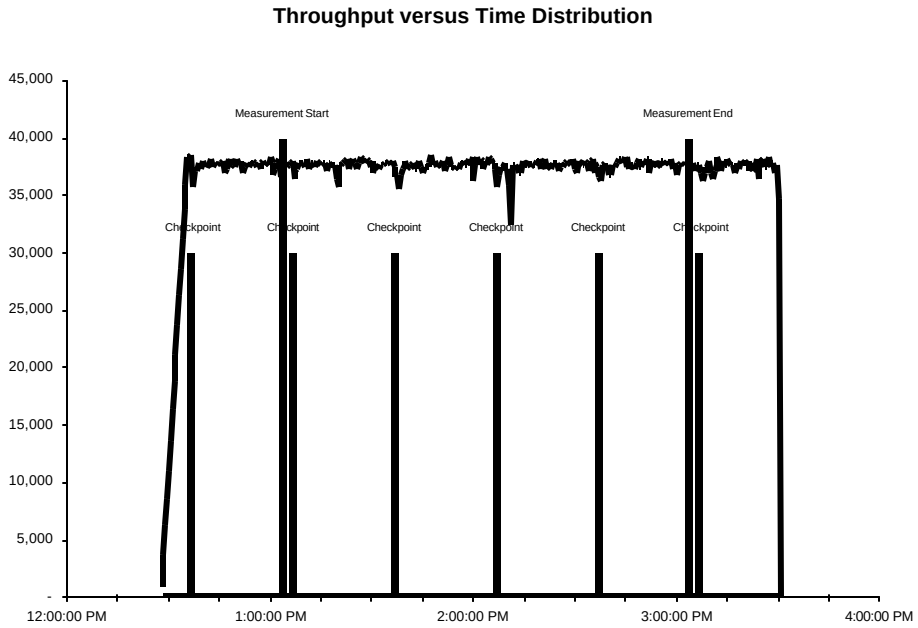


Figure 10: New Order Throughput versus Time

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 checkpoint shortens the time it takes to recover, since the database pages are current and there are no transactions that need to be rolled forward.

5.6.3 Checkpoint Implementation

To perform checkpoints at specific intervals, the SQL Server *recovery interval* was set to 56 and a NT command script was issued to schedule multiple checkpoints at specific 30 minute intervals. The manual checkpoints were spread across 25% of the recovery interval, for a duration of 14 minutes each. During the ramp -up and after all users were active, a background process slept and issued the checkpoint every 30 minutes. 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 checkpoints was verified and are depicted on the graph in Figure 10.

5.6.4 Checkpoint Duration

The start time and duration in seconds of at least the four (4) longest checkpoints during the Measurement Interval must be disclosed.

The checkpoints are detailed in Table 7.

Table 7: Checkpoint Durations

Start Time	Duration
1:36:48 PM	840s
2:06:48 PM	814s
2:36:48 PM	818s
3:06:48 PM	810s

5.7 Measurement Period Duration

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

The measurement interval was 120 minutes.

5.8 Regulation of Transaction Mix

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

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

5.9 Transaction Mix

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

Table 8: Transaction Mix

Type	Percentage
New Order	44.91%
Payment	43.03%
Delivery	4.01%
Stock Level	4.02%
Order Status	4.03%

5.10 Transaction Statistics

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

Table 1contains the required items.

5.11 Checkpoint Count and Location

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

The measurement interval is 120 minutes. There are four checkpoints within the measurement interval. The first checkpoint starts 198 seconds into the measurement interval. The checkpoint interval (time between starts of two consecutive checkpoints) is also 30 minutes.

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 represents 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.
- Giganet clan (1.2Gbit/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:	37,596.34 tpmC
Price per tpmC:	$8.87
Availability:	August 23rd, 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 for 6 processors.
- 1 Microsoft Windows 2000 Advanced Server license.
- 3 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 Lorna Livingtree.

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

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

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

PERFORMANCE METRICS INC.
TPC Certified Auditors

August 21, 2001

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

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

Platform: LH6000
Database Manager: Microsoft SQL Server 2000 Enterprise Edition
Operating System: Microsoft Windows 2000 Advanced Server
Transaction Monitor: Microsoft COM+

Servers: LH6000 with:				
CPU's	Memory	Disks (total)	90% Response	TpmC
6 Pentium III Xeon@700Mhz	Main: 8192 MB Cache: 2 MB	156 @ 18GB 1 @ 9.1 GB	1.37 sec	37,596.34
3 Clients: E800 each with:				
Pentium III Xeon @ 866 Mhz	Main: 512 MB Cache: 256K	1 @ 9.1GB	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 Street, Suite 101, Folsom, CA 95630
(916) 985-1131 fax: (916) 985-1185 email: Lorna@PerfMetrics.com

- The database was properly scaled with 3036 warehouses.
- The ACID properties were successfully demonstrated.
- The ACID properties, except for system loss, were demonstrated on a subset of the SUT configured with a database properly populated for 500 warehouses.
- Input data was generated according to the specified percentages.
- Eight hours of mirrored log space was present on the tested system.
- Eight hours of growth space for the dynamic tables was present on the tested system.
- The data for the 60 day space calculation was verified.
- The controller cache was disabled on the log disk controllers.
- The steady state portion of the test was 120 minutes.
- One checkpoint was taken before the measured interval.
- Four checkpoints were taken during the measured interval.
- The system pricing was checked for major components and maintenance.
- Third party quotes were verified for compliance.

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

/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
 *                      4.21.000 - fixed bug: ~CBaseErr needed to be declared virtual
 */

#pragma once

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

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

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

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

#define ERR_TYPE_LOGIC
        -1              //logic error in program; internal error
#define ERR_SUCCESS
        0               //success (a non-error error)
#define ERR_BAD_ITEM_ID
        1               //expected abort record in txnRecord
#define ERR_TYPE_DELIVERY_POST                                  2
        //expected delivery post failed
#define     ERR_TYPE_WEBDLL
        3               //tpcc web generated error
#define     ERR_TYPE_SQL
        4               //sql server generated error
#define     ERR_TYPE_DBLIB
        5               //dblib generated error
#define     ERR_TYPE_ODBC
        6               //odbc generated error
#define ERR_TYPE_SOCKET
        7               //error on communication socket client rte only
#define ERR_TYPE_DEADLOCK                                       8
                //dblib and odbc only deadlock condition
#define ERR_TYPE_COM                                            9
                //error from COM call
#define ERR_TYPE_TUXEDO
        10              //tuxedo error
#define ERR_TYPE_OS
        11              //operating system error
#define ERR_TYPE_MEMORY
        12              //memory allocation error
#define ERR_TYPE_TPCC_ODBC                                      13
                //error from tpcc odbc txn module
#define ERR_TYPE_TPCC_DBLIB                                     14
                //error from tpcc dblib txn module
#define ERR_TYPE_DELISRV                                        15
                //delivery server error
#define ERR_TYPE_TXNLOG
        16              //txn log error
#define ERR_TYPE_BCCONN
        17              //Benchcraft connection class
#define ERR_TYPE_TPCC_CONN                                      18
                //Benchcraft connection class
#define ERR_TYPE_ENCINA
        19              //Encina error
#define ERR_TYPE_COMPONENT                                      20
                //error from COM component
#define ERR_TYPE_RTE                                            21
                //Benchcraft rte
#define ERR_TYPE_AUTOMATION                                     22
                //Benchcraft automation errors


class CBaseErr
{
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

        virtual int ErrorType() = 0;    // a value which distinguishes the kind of error that occurred
        virtual int ErrorNum() = 0;             // an error value specific to the error type
        virtual char *ErrorText() = 0;          // a string (i.e., human readable) representation of the error

};

class CSocketErr : public CBaseErr
{
public:
        enum Action
        {
                eNone,
                eSend,
                eSocket,
```

```
                eConnect
        };

        CSocketErr(Action eAction, LPCTSTR szLocation);
        CSocketErr(int iError) { m_errId = iError; };
        int             m_errId;
        Action          m_eAction;

        int ErrorType() { return ERR_TYPE_SOCKET;};
        int ErrorNum() { return m_errId;};
        char *ErrorText(void);
};

class CSystemErr : public CBaseErr
{
public:
        enum Action
        {
                eNone,
                eTransactNamedPipe,
                eWaitNamedPipe,
                eSetNamedPipeHandleState,
                eCreateFile,
                eCreateProcess,
                eCallNamedPipe,
                eCreateEvent,
                eCreateThread,
                eVirtualAlloc,
                eReadFile,
                eWriteFile,
                eMapViewOfFile,
                eCreateFileMapping,
                eInitializeSecurityDescriptor,
                eSetSecurityDescriptorDacl,
                eCreateNamedPipe,
                eConnectNamedPipe,
                eWaitForSingleObject,
                eRegOpenKeyEx,
                eRegQueryValueEx,
        };

        CSystemErr(Action eAction, LPCTSTR szLocation);

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

        int             m_errId;
        Action          m_eAction;

        int ErrorType() { return ERR_TYPE_OS;}
        int ErrorNum() { return m_errId;}
        char *ErrorText() { return m_szMsg; }
};

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

        int ErrorType() { return ERR_TYPE_MEMORY;}
        int ErrorNum() { return 0;}
        char *ErrorText() { return "Insufficient Memory to continue.";}
};
```

/common/src/ReadRegistry.cpp

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


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

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

        // determine database protocol to use; may be either ODBC or DBLIB
```

```
		pReg->eDB_Protocol = Unspecified;
		size = sizeof(szTmp);
		if ( RegQueryValueEx(hKey, "DB_Protocol", 0, &type, (BYTE *)&szTmp, &size) ==
ERROR_SUCCESS )
		{
			if ( !stricmp(szTmp, szDBNames[ODBC]) )
				pReg->eDB_Protocol = ODBC;
			else if ( !stricmp(szTmp, szDBNames[DBLIB]) )
				pReg->eDB_Protocol = DBLIB;
		}

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

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

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

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

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

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

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

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

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

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

		RegCloseKey(hKey);

		return FALSE;
}
```

/common/src/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 );
```

/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."
};

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

APPENDIX A - APPLICATION SOURCE CODE

```
        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;
};
```

/common/txnlog/include/rtetime.h

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

#define         MAX_JULIAN_TIME                 0x7FFFFFFFFFFFFFFF
#define JULIAN_TIME     __int64
#define         TC_TIME                 DWORD
extern "C"
{
BOOL                    InitJulianTime(LPSYSTEMTIME lpInitTime);
JULIAN_TIME             GetJulianTime(void);
DWORD                   MyTickCount(void);
void                    GetJulianAndTC(JULIAN_TIME *pJulian, DWORD *pTC);
JULIAN_TIME             ConvertTo64BitTime(int iYear, int iMonth, int iDay, int iHour, int iMinute, int iSecond);
JULIAN_TIME             Get64BitTime(LPSYSTEMTIME lpInitTime);
int                             JulianDay( int yr, int mm, int dd );
void                    JulianToTime(JULIAN_TIME julianTS, int* yr, int* mm, int* dd, int *hh, int *mi, int *ss );
void                    JulianToCalendar( int day, int* yr, int* mm, int* dd );
```

```
}
```

/common/txnlog/include/spinlock.h

```
/*              FILE: SPINLOCK.H
 *
 * Copyright 1997 Microsoft Corp., All rights reserved.
 *
 * Authors: Mike Parkes, Charles Levine, Philip Durr
 *                                      Microsoft Corp.
 */

#ifndef _INC_Spinlock

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

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

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

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

                public:
                        // Public functions.

                        Spinlock( void );

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

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

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

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

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

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

```
                return TRUE;
        }

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

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

        #define _INC_Spinlock

#endif
```

/common/txnlog/include/txnlog.h

```
/*      FILE:                   TXNLOG.H
 *                                      Microsoft TPC-C Kit Ver. 4.10.000
 *
 *                                      NOTE: this file is RTE specific and should not be included
 *                                      in Full Disclosure Reports.
 *
 *                                      Copyright Microsoft, 1999
 *
 *      PURPOSE:    Structure definitions for logging delivery txn completion stats.
 *      Contact:    Charles Levine (clevine@microsoft.com)
 */


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

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

  typedef struct _TXN_ORDERSTATUS
        {
                BYTE        CustByName;
        } TXN_ORDERSTATUS;

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


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

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


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

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


        // TPC-C Txn Record Layout:
        //
  //'TxnStartT0' is a Julian timestamp corresponding to the moment the
  //txn is sent to the SUT, i.e., beginning of response time.  Deltas
  //are in milliseconds.  Note that if RTDelay > 0, then the txn was
  //delayed by this amount.  The delay occurs at the beginning of the
  //response time.  So if RTDelay > 0, then the txn was actually sent
  //at TxnStartT0 + RTDelay.
```

APPENDIX A - APPLICATION SOURCE CODE

```
//
//Graphically:
//
//  time -->
//
//   |--- Menu  ---|-- Keying  --|-- Response --|--- Think  --|
//    <- DeltaT1 -> <- DeltaT2 -> <- DeltaT4  -> <- DeltaT3 ->
//                  ^
//                  ^ TxnStartT0
//
//RTDelay is the amount of response time delay included in DeltaT4.
//RTDelay is recorded per txn because this value can be changed on
//the fly, and so may vary from txn to txn.
//
//TxnStatus is the txn completion code.  It is used to indicate errors.
//For example, in the New Order txn, 1% of txns abort.  TxnStatus will
//reflect this.

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

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

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

                int     DeltaT4;                // response time (ms)
                int                     DeltaTxnExec;   // execution time (ms)
                WORD    w_id;                   // warehouse ID
                BYTE    TxnStatus;              // completion status for txn to indicate errors
                BYTE    reserved;               // for word alignment
                short   o_carrier_id;           // carrier id
                long    o_id[10];               // returned delivery transaction ids
        } TXN_RECORD_TPCC_DELIV_DEF, *PTXN_RECORD_TPCC_DELIV_DEF;



#define         TXN_LOG_VERSION         1
#define         TXN_DATA_START          4096    // offset in log file where log
records start
#define         TXN_LOG_EYE_CATCHER     "BC"    // signature bytes at the start of
log file


        ////////////////////////////////////////////////////////////
        // The transaction log has a header as the first 4K block.
        //
        typedef struct _TXN_LOG_HEADER
        {
                char                    EyeCatcher[2];          // signature bytes;
should always be "BC"
                int                             LogVersion;             // set
to TXN_LOG_VERSION
                JULIAN_TIME             BeginTxnTS;             // timestamp of first
(lowest) txn start
                JULIAN_TIME             EndTxnTS;               // timestamp of last
(highest) txn completion time
                int                             iRecCount;              //
number of records in log file
                BOOL                    bLogSorted;
                int                             iFileSize;              // file
size in bytes

                // the record map provides a fast way to get close to a particular timestamp in a sorted
log file.
//              struct
//              {
//                      JULIAN_TIME             TS;
        // timestamp of record
//                      int                             iPos;
        // byte position in file
//              }                               RecMap[RecMapSize];
//#define       RecMapSize              200
```

```
        } TXN_LOG_HEADER, *PTXN_LOG_HEADER;



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

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

// flags passed in to the constructor
#define     TXN_LOG_WRITE               0x01
#define     TXN_LOG_READ                0x02
#define     TXN_LOG_SORTED              0x04

#define     TXN_LOG_OS_ERROR    1
#define     TXN_LOG_NOT_SORTED          2

#define SKIP_CTRL_RECS                  1

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

                // A "save point" is remembered each time GetNextRecord is called with a start time
specified.
                // The next time it is called, if start time is after the save point, we start scanning from
the
                // save point.  This is particularly useful in FindBestInterval, where the log is scanned
repeatedly.
                JULIAN_TIME                     SavePtTime;
                int                                     iSavePtFilePointer;
                int                                     iSavePtNextRec;
                JULIAN_TIME             lastTS;
        //when writing sorted output, used to verify records are sorted
                BOOL                    bWrite;
        //writing log file

                BOOL                            bLogSorted;                     // is
log file sorted?  applies to both input and output
                JULIAN_TIME                     BeginTxnTS;                     //
timestamp of first (lowest) txn start
                JULIAN_TIME                     EndTxnTS;                       //
timestamp of last (highest) txn completion time
                int                                             iRecCount;
        // number of records in log file

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

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

                DWORD                   dwError;
                HANDLE                  hTxnFile;
        //handle to log file
                HANDLE                  hMapFile;                               //map
file used when sorting the log
                HANDLE                  hIoComplete;            //event to signify
that there are no pending IOs
                HANDLE                  hLogFileIo;
        //event to signal the IO thread to write the inactive buffer

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

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

        public:
                CTxnLog::CTxnLog(LPCTSTR szFileName, DWORD dwOpts);
                ~CTxnLog(void);

                int WriteToLog(PTXN_RECORD_TPCC pTxnRcrd);
                int WriteToLog(PTXN_RECORD_TPCC_DELIV_DEF pTxnRcrd);
                int WriteToLog(PTXN_RECORD_CONTROL pCtrlRec);
                int WriteToLog(PTXN_RECORD_HEADER pCtrlRec);

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

                void CloseTransactionLogFile(void);

                PTXN_RECORD_HEADER GetNextRecord(BOOL bSkipCtrlRecs = FALSE);
```

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

                        int Sort(void);
                        PTXN_RECORD_HEADER GetSortedRecord(int index);

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

class CTXNLOG_ERR : public CBaseErr
{
        public:
                        enum CTPCC_DBLIB_ERRS
                        {
                                ERR_BAD_FILE_FORMAT = 1,        // "File format is invalid."
                                ERR_UNKNOWN_LOG_VERSION,        // "Log file version is unknown."
                                ERR_BROKEN_LOG_FILE,                    // "Log file is
broken."
                                ERR_LOG_NOT_SORTED,                             // "Log
file is not sorted"
                                ERR_INVALID_TIME_SEQ,                   // "Internal Error:
Record Time Sequence invalid."
                        };

                        CTXNLOG_ERR( int iErr ){ m_errno = iErr; };

                        int                     m_errno;

                        int ErrorType() {return ERR_TYPE_TXNLOG;};
                        int ErrorNum() {return m_errno;};

                        // TODO: need to complete...
                        char *ErrorText() {return "";};
};
```

/db_dblib_dll/src/tpcc_dblib.cpp

```
/*      FILE:           TPCC_DBLIB.CPP
 *                      Microsoft TPC-C Kit Ver. 4.20.000
 *                      Copyright Microsoft, 1999
 *      All Rights Reserved
 *
 *                      Version 4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99
 *      PURPOSE:    Implements dblib calls for TPC-C txns.
 *      Contact:    Charles Levine (clevine@microsoft.com)
 *
 * Change history:
 *                  4.20.000 - updated rev number to match kit
 *                  4.10.001 - not deleting error class in catch handler on deadlock retry;
 *                                  not a functional bug, but a memory leak
 *                                  - had to tweak some declarations to compile with latest
SDK; no functional change
 */

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

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

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

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

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

#define     DEFCLPACKSIZE                   4096

// version string; must match return value from tpcc_version stored proc
const char  sVersion[] = "4.10.000";

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

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

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

                case DLL_PROCESS_DETACH:
```

```
                dbexit();                       // close all dblib structures/connections
                break;

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


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

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

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

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

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

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

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

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

    return 0;
}

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

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

    return;
}

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
 *
 */

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

    static SERRORMSG errorMsgs[] =
    {
```

APPENDIX A - APPLICATION SOURCE CODE

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

                                                        }
        };

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

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

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


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

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

        m_MaxRetries = 10;                      // how many retries on deadlock

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

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

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

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


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

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

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

        m_dbproc = dbopen(login, szServer);

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

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

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

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

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

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

		DiscardNextResults(2);

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

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

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

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

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


		DiscardNextRows(0);
		DiscardNextResults(0);
}


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

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

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

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

}

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

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

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

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

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

		CDBLIBERR *pDbLibErr;
		if (m_DbLibErr == NULL)
```

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

                throw pDbLibErr;
}

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

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

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

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

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

                        DiscardNextRows(-1);
                        iResultsRead++;
                }

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

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

                ResetError();

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

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

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

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

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

                                if (pData=dbdata(m_dbproc, 1))
```

```
                                        m_txn.StockLevel.low_stock = *((long *) pData);

                                DiscardNextRows(0);
                                DiscardNextResults(0);

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

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


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

                int                             iTryCount = 0;
                const BYTE    *pData;

                ResetError();

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

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

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

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

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


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

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

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

                                        if(pData=dbdata(m_dbproc, 1))
                                                UtilStrCpy(m_txn.NewOrder.OL[i].ol_i_name,
pData, dbdatlen(m_dbproc, 1));
                                        if(pData=dbdata(m_dbproc, 2))
```

APPENDIX A - APPLICATION SOURCE CODE

```
                                        m_txn.NewOrder.OL[i].ol_stock =
(*(DBSMALLINT *) pData);
                                if(pData=dbdata(m_dbproc, 3))

        UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_generic, pData, dbdatlen(m_dbproc, 3));
                                if(pData=dbdata(m_dbproc, 4))
                                        dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,4),
                                        SQLFLT8, (BYTE
*)&m_txn.NewOrder.OL[i].ol_i_price, 8);
                                if(pData=dbdata(m_dbproc, 5))

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

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

                                        DiscardNextRows(0);
                        }


                        // get remaining values for w_tax, d_tax, o_id, c_last, c_discount, c_credit,
o_entry_d, commit_flag
                        if (dbresults(m_dbproc) != SUCCEED)
                                ThrowError(CDBLIBERR::eDbResults);

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

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

                        if (pData=dbdata(m_dbproc, 1))


                                dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,1), SQLFLT8, (BYTE *)&m_txn.NewOrder.w_tax, 8);
                        if (pData=dbdata(m_dbproc, 2))

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

                        DiscardNextRows(0);
                        DiscardNextResults(0);


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

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

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

APPENDIX A - APPLICATION SOURCE CODE

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

        int                     iTryCount = 0;
        const BYTE    *pData;

        ResetError();

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

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

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

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

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

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

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

                        if (pData=dbdata(m_dbproc, 1))
                                m_txn.Payment.c_id = *((DBINT *) pData);
                        if (pData=dbdata(m_dbproc, 2))
                                UtilStrCpy(m_txn.Payment.c_last, pData,
dbdatlen(m_dbproc, 2));
                        if (pData=dbdata(m_dbproc, 3))
                        {
                                datetime = *((DBDATETIME *) pData);
                                dbdatecrack(m_dbproc, &daterec, &datetime);
                                m_txn.Payment.h_date.year   = daterec.year;
                                m_txn.Payment.h_date.month  = daterec.month;
                                m_txn.Payment.h_date.day    = daterec.day;
                                m_txn.Payment.h_date.hour   = daterec.hour;
                                m_txn.Payment.h_date.minute = daterec.minute;
                                m_txn.Payment.h_date.second = daterec.second;
                        }
                        if (pData=dbdata(m_dbproc, 4))
                                UtilStrCpy(m_txn.Payment.w_street_1, pData,
dbdatlen(m_dbproc, 4));
                        if (pData=dbdata(m_dbproc, 5))
                                UtilStrCpy(m_txn.Payment.w_street_2, pData,
dbdatlen(m_dbproc, 5));
                        if (pData=dbdata(m_dbproc, 6))
                                UtilStrCpy(m_txn.Payment.w_city, pData,
dbdatlen(m_dbproc, 6));
                        if (pData=dbdata(m_dbproc, 7))
                                UtilStrCpy(m_txn.Payment.w_state, pData,
dbdatlen(m_dbproc, 7));
                        if (pData=dbdata(m_dbproc, 8))
                                UtilStrCpy(m_txn.Payment.w_zip, pData,
dbdatlen(m_dbproc, 8));
                        if (pData=dbdata(m_dbproc, 9))
                                UtilStrCpy(m_txn.Payment.d_street_1, pData,
dbdatlen(m_dbproc, 9));
                        if (pData=dbdata(m_dbproc, 10))
                                UtilStrCpy(m_txn.Payment.d_street_2, pData,
dbdatlen(m_dbproc, 10));
                        if (pData=dbdata(m_dbproc, 11))
                                UtilStrCpy(m_txn.Payment.d_city, pData,
dbdatlen(m_dbproc, 11));
                        if (pData=dbdata(m_dbproc, 12))
                                UtilStrCpy(m_txn.Payment.d_state, pData,
dbdatlen(m_dbproc, 12));
                        if (pData=dbdata(m_dbproc, 13))
                                UtilStrCpy(m_txn.Payment.d_zip, pData,
dbdatlen(m_dbproc, 13));
                        if ( pData=dbdata(m_dbproc, 14))
                                UtilStrCpy(m_txn.Payment.c_first, pData,
dbdatlen(m_dbproc, 14));
                        if (pData=dbdata(m_dbproc, 15))
                                UtilStrCpy(m_txn.Payment.c_middle, pData,
dbdatlen(m_dbproc, 15));
                        if (pData=dbdata(m_dbproc, 16))
                                UtilStrCpy(m_txn.Payment.c_street_1, pData,
dbdatlen(m_dbproc, 16));
```

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

                DiscardNextRows(0);
                DiscardNextResults(0);


                if (m_txn.Payment.c_id == 0)
                        throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
                else
                        m_txn.Payment.exec_status_code = eOK;
                        return;
                }
                catch (CSQLERR *e)
                {
                        if ((e->m_msgno == 1205 ||
                                (e->m_msgno == iErrOleDbProvider &&
                                strstr(e->m_msgtext, sErrTimeoutExpired) != NULL)) &&
                                (++iTryCount <= iMaxRetries))
                        {
                                // hit deadlock; backoff for increasingly longer period
                                delete e;
                                Sleep(10 * iTryCount);
                        }
                        else
                                throw;
                }
        }       // while (TRUE)

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



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

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

        ResetError();

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

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

                        // if customer id is zero, then order status is by name
                        if (m_txn.OrderStatus.c_id == 0)
```

```
                                        dbrpcparam(m_dbproc, NULL, 0, SQLCHAR, -1,
strlen(m_txn.OrderStatus.c_last), (unsigned char *)m_txn.OrderStatus.c_last);


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

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

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

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

                                        if(pData=dbdata(m_dbproc, 1))
                                                m_txn.OrderStatus.OL[i].ol_supply_w_id =
(*(DBSMALLINT *) pData) ;
                                        if(pData=dbdata(m_dbproc, 2))
                                                m_txn.OrderStatus.OL[i].ol_i_id = (*(DBINT *)
pData);
                                        if(pData=dbdata(m_dbproc, 3))
                                                m_txn.OrderStatus.OL[i].ol_quantity =
(*(DBSMALLINT *) pData);
                                        if(pData=dbdata(m_dbproc, 4))
                                                dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,4),
                                                                        SQLFLT8, (BYTE
*)&m_txn.OrderStatus.OL[i].ol_amount, 8);
                                        if(pData=dbdata(m_dbproc, 5))
                                        {
                                                datetime = *((DBDATETIME *) pData);
                                                dbdatecrack(m_dbproc, &daterec, &datetime);
                                                m_txn.OrderStatus.OL[i].ol_delivery_d.year   =
daterec.year;
                                                m_txn.OrderStatus.OL[i].ol_delivery_d.month
= daterec.month;
                                                m_txn.OrderStatus.OL[i].ol_delivery_d.day    =
daterec.day;
                                                m_txn.OrderStatus.OL[i].ol_delivery_d.hour   =
daterec.hour;
                                                m_txn.OrderStatus.OL[i].ol_delivery_d.minute
= daterec.minute;
                                                m_txn.OrderStatus.OL[i].ol_delivery_d.second
= daterec.second;
                                        }
                                        i++;
                                }
                                m_txn.OrderStatus.o_ol_cnt = i;


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

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

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

                                if(pData=dbdata(m_dbproc, 1))
                                        m_txn.OrderStatus.c_id = (*(DBINT *) pData);
                                if(pData=dbdata(m_dbproc, 2))
                                        UtilStrCpy(m_txn.OrderStatus.c_last, pData,
dbdatlen(m_dbproc,2));
                                if(pData=dbdata(m_dbproc, 3))
                                        UtilStrCpy(m_txn.OrderStatus.c_first, pData,
dbdatlen(m_dbproc,3));
                                if(pData=dbdata(m_dbproc, 4))
                                        UtilStrCpy(m_txn.OrderStatus.c_middle, pData,
dbdatlen(m_dbproc, 4));
                                if(pData=dbdata(m_dbproc, 5))
                                {
                                        datetime = *((DBDATETIME *) pData);
                                        dbdatecrack(m_dbproc, &daterec, &datetime);
                                        m_txn.OrderStatus.o_entry_d.year   = daterec.year;
                                        m_txn.OrderStatus.o_entry_d.month  = daterec.month;
                                        m_txn.OrderStatus.o_entry_d.day    = daterec.day;
                                        m_txn.OrderStatus.o_entry_d.hour   = daterec.hour;
                                        m_txn.OrderStatus.o_entry_d.minute = daterec.minute;
                                        m_txn.OrderStatus.o_entry_d.second = daterec.second;
                                }
                                if(pData=dbdata(m_dbproc, 6))
                                        m_txn.OrderStatus.o_carrier_id = (*(DBSMALLINT *) pData);
                                if(pData=dbdata(m_dbproc, 7))
                                        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,7),
                                                                        SQLFLT8, (BYTE
*)&m_txn.OrderStatus.c_balance, 8);
```

```
                        if(pData=dbdata(m_dbproc, 8))
                                m_txn.OrderStatus.o_id = (*(DBINT *) pData);

                        DiscardNextRows(0);
                        DiscardNextResults(0);


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

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

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


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

        ResetError();

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

                        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&m_txn.Delivery.w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.Delivery.o_carrier_id);

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

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

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

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

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

                        DiscardNextRows(0);
                        DiscardNextResults(0);

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

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

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

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

/db_dblib_dll/src/tpcc_dblib.h

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

#ifndef PDBPROCESS
#define DBPROCESS void   // dbprocess structure type
typedef DBPROCESS * PDBPROCESS;
#endif

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

class CSQLERR : public CBaseErr
{
        public:
                CSQLERR(void)
                {
                        m_msgno = 0;
                        m_msgstate = 0;
                        m_severity = 0;
                        m_msgtext = NULL;
                };

                ~CSQLERR()
                {
                        delete [] m_msgtext;
                };

                int             m_msgno;
                int             m_msgstate;
                int             m_severity;
                char   *m_msgtext;

                int ErrorType() {return ERR_TYPE_SQL;};
                int ErrorNum() {return m_msgno;};
                char *ErrorText() {return m_msgtext;};

};

class CDBLIBERR : public CBaseErr
{
        public:
                enum ACTION
                {
                        eNone,
                        eUnknown,
                        eLogin,                 // error from dblogin
                        eDbOpen,                // error from dbopen
                        eDbUse,                 // error from dbuse
                        eDbSqlExec,             // error from dbsqlexec
                        eDbSet,                 // error from one of
the dbset* routines
                        eDbNextRow,             // error from
dbnextrow
                        eWrongRowCount,         // more or less rows returned than
expected
                        eWrongNumCols,          // more or less columns returned
than expected
                        eDbResults,             // error from dbresults
                        eDbRpcExec,             // error from
dbrpcexec
                        eDbSetMaxProcs,         // error from dbsetmaxprocs
                        eDbProcHandler          // error from either
dbprocerrhandle or dbprocmsghandle
                };

                CDBLIBERR(ACTION eAction, int severity = 0, int dberror = 0, int oserr = 0)
```

```
                {
                        m_eAction = eAction;
                        m_severity = severity;
                        m_dberror = dberror;
                        m_oserr = oserr;

                        m_dberrstr = NULL;
                        m_oserrstr = NULL;
                };

                ~CDBLIBERR()
                {
                        delete [] m_dberrstr;
                        delete [] m_oserrstr;
                };

                ACTION  m_eAction;
                int             m_severity;
                int             m_dberror;
                int             m_oserr;
                char   *m_dberrstr;
                char   *m_oserrstr;

                int ErrorType() {return ERR_TYPE_DBLIB;};
                int ErrorNum() {return m_dberror;};
                char *ErrorText() {return m_dberrstr;};

};

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

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

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

                int             m_errno;
                int             m_iTryCount;

                int ErrorType() {return ERR_TYPE_TPCC_DBLIB;};
                int ErrorNum() {return m_errno;};

                char *ErrorText();
};

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

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

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

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

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

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

```
                        void OrderStatus        ();


                        // these are public because they must be called from the dblib err_handler and
msg_hangler
                        // outside of the class
                        void SetDbLibError(int severity, int dberr, int oserr, LPCSTR dberrstr, LPCSTR oserrstr);
                        void SetSqlError( int msgno, int msgstate, int severity, LPCSTR msgtext );

};

extern "C" DllDecl CTPCC_DBLIB* CTPCC_DBLIB_new
                ( LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword, LPCSTR szHost, LPCSTR szDatabase
);

typedef CTPCC_DBLIB* (TYPE_CTPCC_DBLIB)(LPCSTR, LPCSTR, LPCSTR, LPCSTR, LPCSTR);
```

/isapi_dll/src/resource.h

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

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

/isapi_dll/src/tpcc.cpp

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

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

#include <sqltypes.h>

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

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

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

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

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

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

APPENDIX A - APPLICATION SOURCE CODE

```
#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];

// debugging....
// DebugBreak();

        try
        {
            switch( ul_reason_for_call )
            {
                case DLL_PROCESS_ATTACH:
                    {
                        DWORD dwSize =
MAX_COMPUTERNAME_LENGTH+1;
                        GetComputerName(szMyComputerName,
&dwSize);
                        szMyComputerName[dwSize] = 0;
                    }

                    DisableThreadLibraryCalls((HMODULE)hModule);
```

```
                InitializeCriticalSection(&TermCriticalSection);

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

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

                TermInit();

                // load DLL for txn monitor
                if (Reg.eTxnMon == TUXEDO)
                {
                        strcpy( szDllName, Reg.szPath );
                        strcat( szDllName, "tpcc_tuxedo.dll");
                        hLibInstanceTm = LoadLibrary( szDllName );
                        if (hLibInstanceTm == NULL)
                                throw new CWEBCLNT_ERR(
ERR_LOADDLL_FAILED, szDllName, GetLastError() );

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

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

                // load DLL for database connection
                if ((Reg.eTxnMon == None) || (dwNumDeliveryThreads > 0))
                {
                        if (Reg.eDB_Protocol == DBLIB)
                        {
                                strcpy( szDllName, Reg.szPath );
                                strcat( szDllName,
"tpcc_dblib.dll");
                                hLibInstanceDb = LoadLibrary(
szDllName );
                                if (hLibInstanceDb == NULL)
                                        throw new
CWEBCLNT_ERR( ERR_LOADDLL_FAILED, szDllName, GetLastError() );

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

                                // get function pointer to wrapper
for class constructor
```

APPENDIX A - APPLICATION SOURCE CODE

```
                                                pCTPCC_ODBC_new =
(TYPE_CTPCC_ODBC*) GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
                                                if (pCTPCC_ODBC_new ==
NULL)
                                                        throw new
CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                                }
                                        }

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

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

                                                InitJulianTime(NULL);

                                                // create unique log file name based on delilog -
yymmdd-hhmm.log
                                                SYSTEMTIME Time;
                                                GetLocalTime( &Time );
                                                wsprintf( szLogFile, "%sdelivery-
%2.2d%2.2d%2.2d-%2.2d%2.2d.log",
                                                        Reg.szPath,
Time.wYear % 100, Time.wMonth, Time.wDay, Time.wHour, Time.wMinute );
                                                txnDelilog = new CTxnLog(szLogFile,
TXN_LOG_WRITE);

                                                //write event into txn log for START
                                                txnDelilog -
>WriteCtrlRecToLog(TXN_EVENT_START, szMyComputerName, sizeof(szMyComputerName));

                                                // allocate structures for delivery buffers and
thread mgmt
                                                pDeliHandles = new
HANDLE[dwNumDeliveryThreads];
                                                pDelBuff = new
DELIVERY_TRANSACTION[dwDelBuffSize];
                                                // launch DeliveryWorkerThread to perform
actual delivery txns
                                                for(i=0; i<dwNumDeliveryThreads; i++)
                                                {
                                                        pDeliHandles[i] = (HANDLE)
_beginthread( DeliveryWorkerThread, 0, NULL );
                                                        if (pDeliHandles[i] ==
INVALID_HANDLE_VALUE)
                                                                throw new
CWEBCLNT_ERR( ERR_DELIVERY_THREAD_FAILED );
                                                        }
                                                }

                                                break;

                                        case DLL_PROCESS_DETACH:
                                                if (dwNumDeliveryThreads)
                                                {
                                                        if (txnDelilog != NULL)
                                                        {
                                                                //write event into txn log for STOP
                                                                txnDelilog -
>WriteCtrlRecToLog(TXN_EVENT_STOP, szMyComputerName, sizeof(szMyComputerName));

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

                                                        delete [] pDeliHandles;
                                                        delete [] pDelBuff;

                                                        CloseHandle( hWorkerSemaphore );
                                                        CloseHandle( hDoneEvent );
                                                        DeleteCriticalSection(&DelBuffCriticalSection);
                                                }

                                                DeleteCriticalSection(&TermCriticalSection);

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

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

                                                Sleep(500);
                                                break;

                                        default:
                                                /* nothing */;
                        }
                }
                catch (CBaseErr *e)
                {
                        WriteMessageToEventLog( e->ErrorText() );
```

```
                delete e;
                TerminateExtension(0);
                return FALSE;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception.  DLL could not load."));
                TerminateExtension(0);
                return FALSE;
        }

        return TRUE;
}


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

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

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

        return TRUE;
}

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

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

        TermDeleteAll();
        return TRUE;
}


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

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

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


#ifdef ICECAP
        StartCAP();
#endif

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

                if (TermId != 0)
                {
                        if ( TermId < 0 || TermId >= Term.iNumEntries ||
Term.pClientData[TermId].iNextFree != -1 )
                        {
                                // debugging...
```

APPENDIX A - APPLICATION SOURCE CODE

```
                        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 (...)
    {
```

APPENDIX A - APPLICATION SOURCE CODE

```
                ErrorForm( pECB, ERR_TYPE_WEBDLL, 0, TermId, iSyncId, "Error: Unhandled
exception in Web Client.", szBuffer );
        }

#ifdef ICECAP
        StopCAP();
#endif


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

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

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


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

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

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

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

      (VOID) DeregisterEventSource(hEventSource);
   }
}


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

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

        DELIVERY_TRANSACTION                    delivery;
        PDELIVERY_DATA                          pDeliveryData;
        TXN_RECORD_TPCC_DELIV_DEF       txnDeliRec;

        DWORD                                   index;
        HANDLE                                  handles[2];

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

        assert(txnDelilog != NULL);

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

```
                WriteMessageToEventLog( szTmp );
                delete e;
                goto ErrorExit;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception caught in
DeliveryWorkerThread."));
                goto ErrorExit;
        }

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

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

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

                                LeaveCriticalSection(&DelBuffCriticalSection);

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

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

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

                                GetLocalTime( &trans_end );

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

                                if (txnDelilog != NULL)
                                        txnDelilog->WriteToLog(&txnDeliRec);
                        }
                }
                catch (CBaseErr *e)
                {
                        char szTmp[1024];
                        wsprintf( szTmp, "Error in Delivery Txn thread.  %s", e->ErrorText() );
                        WriteMessageToEventLog( szTmp );

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

                        delete e;
                }
                catch (...)
                {
                        // unhandled exception; shouldn't happen; not much we can do...
                        WriteMessageToEventLog(TEXT("Unhandled exception caught in
DeliveryWorkerThread."));
                }
        }

ErrorExit:
        delete pTxn;
        _endthread();
}

/* FUNCTION: PostDeliveryInfo
 *
 * PURPOSE:             This function enters the delivery txn into the deferred delivery buffer.
 *
 * RETURNS:                     BOOL        FALSE       delivery information posted successfully
 *                                                      TRUE        error cannot post
delivery info
 */

BOOL PostDeliveryInfo(short w_id, short o_carrier_id)
{
```

```
	BOOL bError;

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

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

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

	return bError;
}

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

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

	//allowable client command strings i.e. CMD=command
	static char *szCmds[] =
	{
		"Process", "..NewOrder..", "..Payment..", "..Delivery..", "..Order-Status..", "..Stock-
Level..",
		"..Exit..", "Submit", "Menu", "Clear", "Stats", ""
	};

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

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

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

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

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

/* FUNCTION: void WelcomeForm
 *
 */

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

	//welcome to tpc-c html form buffer, this is first form client sees.
	strcpy( szBuffer, "<HTML><HEAD><TITLE>TPC-C Web
Client</TITLE></HEAD><BODY>"
		"<B><BIG>Microsoft TPC-C Web
Client (ver 4.20)</BIG></B> <BR> <BR>"
		"<font face=\"Courier
New\"><PRE>"
		"Compiled: "__DATE__",
"__TIME__" <BR>"
		"Source:  "__FILE__"
("__TIMESTAMP__") <BR>"
		"</PRE></font>"
```

APPENDIX A - APPLICATION SOURCE CODE

```
                                                                "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
                                                                "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"0\">"
                                                                "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
                                                                "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"1\">"
                                                                "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"0\">"
                                                                "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"0\">"
                                                                "<INPUT TYPE=\"hidden\"
NAME=\"VERSION\" VALUE=\"" WEBCLIENT_VERSION "\">"
                                        );

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

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

        if (Reg.eTxnMon == None)
                // connection options may be specified when not using a txn monitor
                sprintf( szTmp,         "Please enter your database options for this
connection:<BR>"
                                                                "<font face=\"Courier New\"
color=\"blue\"><PRE>"
                                                                "DB Server    = <INPUT
NAME=\"db_server\" SIZE=20 VALUE=\"%s\"><BR>"
                                                                "DB User ID   = <INPUT
NAME=\"db_user\" SIZE=20 VALUE=\"%s\"><BR>"
                                                                "DB Password  = <INPUT
NAME=\"db_passwd\" SIZE=20 VALUE=\"%s\"><BR>"
                                                                "DB Name      = <INPUT
NAME=\"db_name\" SIZE=20 VALUE=\"%s\"><BR>"
                                                                "</PRE></font>"
                                        , Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
Reg.szDbName );
        else
                // if using a txn monitor, connection options are determined from registry; can't
                // set per user.  show options fyi
                sprintf( szTmp,         "Database options which will be used by the transaction
monitor:<BR>"
                                                                "<font face=\"Courier New\"
color=\"blue\"><PRE>"
                                                                "DB Server            =
<B>%s</B><BR>"
                                                                "DB User ID           =
<B>%s</B><BR>"
                                                                "DB Password          =
<B>%s</B><BR>"
                                                                "DB Name              =
<B>%s</B><BR>"
                                                                "</PRE></font>"
                                        , Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
Reg.szDbName );
        strcat( szBuffer, szTmp);

        sprintf( szTmp,         "Please enter your Warehouse and District for this session:<BR>"
                                                        "<font face=\"Courier New\"
color=\"blue\"><PRE>" );
        strcat( szBuffer, szTmp);
        strcat( szBuffer,       "Warehouse ID = <INPUT NAME=\"w_id\" SIZE=4><BR>"
                                                                "District ID  = <INPUT
NAME=\"d_id\" SIZE=2><BR>"
                                                                "</PRE></font><HR>"
                                                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Submit\">"
                                                                "</FORM></BODY></HTML>");
}

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

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

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

        // validate version field; the version field ensures that the RTE is synchronized with the web client
        GetKeyValue(&ptr, "VERSION", szVersion, sizeof(szVersion), ERR_VERSION_MISMATCH);
```

```
        if ( strcmp( szVersion, WEBCLIENT_VERSION ) )
                throw new CWEBCLNT_ERR( ERR_VERSION_MISMATCH );

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

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

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


        iNewTerm = TermAdd();

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

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

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

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

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

        EnterCriticalSection(&TermCriticalSection);

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

        LeaveCriticalSection(&TermCriticalSection);

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

char *CWEBCLNT_ERR::ErrorText()
{
        static SERRORMSG errorMsgs[] =
        {
                {       ERR_COMMAND_UNDEFINED,
                "Command undefined."
                                                                                        },
                {       ERR_D_ID_INVALID,
                "Invalid District ID Must be 1 to 10."
                                        },
                {       ERR_DELIVERY_CARRIER_ID_RANGE,
        "Delivery Carrier ID out of range must be 1 - 10."
        },
                {       ERR_DELIVERY_CARRIER_INVALID,
        "Delivery Carrier ID invalid must be numeric 1 - 10."                   },
```

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

```
		{	ERR_PAYMENT_LAST_NAME_TO_LONG,
	"Payment Customer last name longer than 16 characters."
	},
		{	ERR_PAYMENT_MISSING_CDI_KEY,
	"Payment missing Customer district key \"CDI*\"."
	},
		{	ERR_PAYMENT_MISSING_CID_CLT,
	"Payment Either Customer ID or Last Name must be entered."	},
		{	ERR_PAYMENT_MISSING_CID_KEY,
	"Payment missing Customer Key \"CID*\"."
			},
		{	ERR_PAYMENT_MISSING_CLT_KEY,
	"Payment missing Customer Last Name key \"CLT*\"."
	},
		{	ERR_PAYMENT_MISSING_CWI_KEY,
	"Payment missing Customer Warehouse key \"CWI*\"."
		},
		{	ERR_PAYMENT_MISSING_DID_KEY,
	"Payment missing District Key \"DID*\"."
		},
		{	ERR_PAYMENT_MISSING_HAM_KEY,
	"Payment missing Amount key \"HAM*\"."
				},
		{	ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,	"Stock Level;
missing Threshold key \"TT*\"."	},
		{	ERR_STOCKLEVEL_THRESHOLD_INVALID,	"Stock
Level; Threshold value must be in the range = 1 - 99."		},
		{	ERR_STOCKLEVEL_THRESHOLD_RANGE,
	"Stock Level Threshold out of range, range must be 1 - 99."	},
		{	ERR_VERSION_MISMATCH,
		"Invalid version field.  RTE and Web Client are probably out of sync." },
		{	ERR_W_ID_INVALID,
		"Invalid Warehouse ID."
	},
		{	0,
			""

		}
	};

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

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

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

}

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

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

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

```
        if ( *ptr != '=' )
                goto ErrorExit;
        ptr++;

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

        *pQueryString = ptr;
        return;

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

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

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

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

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

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

        *pQueryString = ptr;
        return atoi(ptr0);

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

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

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

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

        Term.pClientData        = NULL;
        Term.pClientData        = (PCLIENTDATA)malloc(Term.iNumEntries * sizeof(CLIENTDATA));
        if (Term.pClientData == NULL)
```

```
		{
			LeaveCriticalSection(&TermCriticalSection);
			throw new CWEBCLNT_ERR( ERR_MEM_ALLOC_FAILED );
		}

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

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

		LeaveCriticalSection(&TermCriticalSection);
}

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

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

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

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

		LeaveCriticalSection(&TermCriticalSection);
}

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

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

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

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

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

		LeaveCriticalSection(&TermCriticalSection);
		return iNewTerm;
}

/* FUNCTION: TermDelete
 *
 * PURPOSE:		This function makes a terminal entry in the Term array available for reuse.
```

APPENDIX A - APPLICATION SOURCE CODE

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

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

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

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

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

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

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

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

```
                    );
        }
        else
        {
                c += wsprintf(szForm+c, "Warehouse: %4.4d   District: %2.2d                Date: ",
                        pNewOrderData->w_id,
                        pNewOrderData->d_id);

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

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

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

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

                        i = 0;
                }

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

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

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

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

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

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

APPENDIX A - APPLICATION SOURCE CODE

```
                "<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..\">"
```

APPENDIX A - APPLICATION SOURCE CODE

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

}

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

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

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

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

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

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

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

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

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

```
		}
}

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

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

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

		if ( bInput )
		{
			strcpy( szForm+c,
				"Carrier Number: <INPUT NAME=\"OCD*\" SIZE=1><BR> <BR>"
				"Execution Status: <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"
				" <BR> <BR> <BR> <BR> <BR> <BR> <BR> <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) );
```

APPENDIX A - APPLICATION SOURCE CODE

```
	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 fr om 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/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.
 *
 */
```

APPENDIX A - APPLICATION SOURCE CODE

```
//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
```

/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;
        ULONG ulTmpSize = 0;

        m_pTxn                          = NULL;
        m_pNewOrder                     = NULL;
```

```
	m_pPayment		= NULL;
	m_pStockLevel		= NULL;
	m_pOrderStatus		= NULL;

	m_bSinglePool		= bSinglePool;

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

void CTPCC_COM::NewOrder()
{
	VARIANT	vTxn_out;

	HRESULT hr = m_pNewOrder->NewOrder(m_vTxn, &vTxn_out);
	if (FAILED(hr))
		throw new CCOMERR( hr );
```

```
		memcpy(m_pTxn, (void *)vTxn_out.parray->pvData,vTxn_out.parray->rgsabound[0].cElements);
		SafeArrayDestroy(vTxn_out.parray);

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

void CTPCC_COM::Payment()
{
		VARIANT	vTxn_out;

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

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

void CTPCC_COM::StockLevel()
{
		VARIANT	vTxn_out;

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

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

void CTPCC_COM::OrderStatus()
{
		VARIANT	vTxn_out;

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

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

/tm_com_dll/src/tpcc_com.h

```
/*		FILE:			TPCC_COM.H
 *
 *
 *	All Rights Reserved
 *
 *
 *
 *		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;

                        VARIANT m_vTxn;
        public:
                        CTPCC_COM(BOOL bSinglePool);
                        ~CTPCC_COM(void);
                                inline PNEW_ORDER_DATA                  BuffAddr_NewOrder()
                { return &m_pTxn->u.NewOrder;   };
                                inline PPAYMENT_DATA            BuffAddr_Payment()              {
return &m_pTxn->u.Payment;    };
                                inline PDELIVERY_DATA           BuffAddr_Delivery()             {
return &m_pTxn->u.Delivery;   };
                                inline PSTOCK_LEVEL_DATA        BuffAddr_StockLevel()   { return &m_pTxn-
>u.StockLevel;  };
                                inline PORDER_STATUS_DATA       BuffAddr_OrderStatus()  { return &m_pTxn-
>u.OrderStatus; };

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


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


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

typedef CTPCC_COM* (TYPE_CTPCC_COM)(BOOL);
```

/tpcc_com_ps/src/tpcc_com_ps.def

```
LIBRARY     "tpcc_com_ps"

DESCRIPTION  'Proxy/Stub DLL'

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

APPENDIX A - APPLICATION SOURCE CODE

/tpcc_com_ps/src/tpcc_com_ps.h

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

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


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


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

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

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

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

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

/* Forward Declarations */

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


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

#ifdef __cplusplus
extern "C"{
#endif

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

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



extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

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


EXTERN_C const IID IID_ITPCC;

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

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

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

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

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

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

```
    virtual HRESULT __stdcall CallSetComplete( void) = 0;

  };

#else           /* C style interface */

  typedef struct ITPCCVtbl
  {
    BEGIN_INTERFACE

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

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

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

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

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

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

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

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

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

    END_INTERFACE
  } ITPCCVtbl;

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


#ifdef COBJMACROS


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

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

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


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

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

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

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

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

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

#endif /* COBJMACROS */


#endif          /* C style interface */


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


void __RPC_STUB ITPCC_NewOrder_Stub(
```

```
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


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


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


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


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


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


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


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


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


HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC __RPC_FAR * This);


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



#endif          /* __ITPCC_INTERFACE_DEFINED__ */


/* Additional Prototypes for ALL interfaces */

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

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

/tpcc_com_ps/src/tpcc_com_ps.idl

```
/*          FILE:                   ITPCC.IDL
 *                                              Microsoft TPC-C Kit Ver. 4.20.000
 *                                              Copyright Microsoft, 1999
 *      All Rights Reserved
```

APPENDIX A - APPLICATION SOURCE CODE

```
*                                                                                                                                   [out] VARIANT *txn_out
*                                                  not yet audited                                                                  );
*
*           PURPOSE:   Defines the interface used by TPCC. This interface can be implemented by C++          HRESULT _stdcall CallSetComplete
components.
*                                                                                                                                   (
*                                                                                                                                   );
*  Change history:
*                          4.20.000 - first version                                                  };          // interface ITPCC
*/

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

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

    HRESULT _stdcall NewOrder
                                                        (
                                                        [in] VARIANT  txn_in,
                                                        [out] VARIANT *txn_out
                                                        );

    HRESULT _stdcall Payment
                                                        (
                                                        [in] VARIANT txn_in,
                                                        [out] VARIANT *txn_out
                                                        );

    HRESULT _stdcall Delivery
                                                        (
                                                        [in] VARIANT txn_in,
                                                        [out] VARIANT *txn_out
                                                        );

    HRESULT _stdcall StockLevel
                                                        (
                                                        [in] VARIANT txn_in,
                                                        [out] VARIANT *txn_out
                                                        );

    HRESULT _stdcall OrderStatus
                                                        (
                                                        [in] VARIANT txn_in,
```

Appendix B Database Design

B.1 Create, backup and restore

SETUP.CMD

```
::@ECHO OFF

@ECHO ********************************************************************************
@ECHO *                                                                              *
@ECHO *  Microsoft TPC-C V3 Benchmark Kit Ver. 4.22                                  *
@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

: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\bulkload.log
@if errorlevel 1 goto DBOPT1_ERROR
@ECHO Beginning data load and index creation...
@if '%4'==''      loader\bin\tpccldr -S%1 -W%2 -flogs\bulkload.log -dscripts\%2.war\ddl -c0
@if errorlevel 1 goto END
@if '%4'=='normal' loader\bin\tpccldr -S%1 -W%2 -flogs\bulkload.log -dscripts\%2.war\ddl -c0
@if errorlevel 1 goto END
@if '%4'=='scale_down' loader\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...
@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.

@ECHO.
@ECHO Calculating initial database space usage...
@cd..\acid\space
@call space.cmd %1
@cd..\..\setup

@if  '%3'=='full' goto backup
```

APPENDIX B – DATABASE DESIGN

```
@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

:end

echo on
```

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                        = MSSQL_tpcc_root,
            FILENAME    = "C:\MSSQL_tpcc_root.mdf",
            SIZE                        = 10MB,
            FILEGROWTH                  = 0),
FILEGROUP MSSQL_misc_fg
(           NAME                        = MSSQL_misc1,
            FILENAME    = "F:",
            SIZE                        = 29000MB,
            FILEGROWTH                  = 0),
(           NAME                        = MSSQL_misc2,
            FILENAME    = "G:",
            SIZE                        = 29000MB,
            FILEGROWTH                  = 0),
(           NAME                        = MSSQL_misc3,
            FILENAME    = "H:",
            SIZE                        = 29000MB,
            FILEGROWTH                  = 0),
FILEGROUP MSSQL_cs_fg
(           NAME                        = MSSQL_cs1,
            FILENAME    = "M:",
            SIZE                        = 60000MB,
            FILEGROWTH                  = 0),
(           NAME                        = MSSQL_cs2,
            FILENAME    = "N:",
            SIZE                        = 60000MB,
            FILEGROWTH                  = 0),
(           NAME                        = MSSQL_cs3,
            FILENAME    = "O:",
            SIZE                        = 60000MB,
            FILEGROWTH                  = 0)
LOG ON
(           NAME                        =MSSQL_tpcc_log,
            FILENAME    ="E:",
            SIZE                        =101400MB,
            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
```

BACKUP.SQL

```
-- File:    BACKUP.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates backup of tpcc database


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

dump database tpcc to tpccback1, tpccback2, tpccback3, tpccback4 with init, stats = 1

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

go
```

BACKUPDEV.SQL

```
-- File:    BACKUPDEVB.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates tpcc database Backup Devices


use master
go

--  create backup devices

exec sp_addumpdevice 'disk','tpccback1','v:\tpccback1.dmp'
go
```

```
exec sp_addumpdevice 'disk','tpccback2','v:\tpccback2.dmp'
go
exec sp_addumpdevice 'disk','tpccback3','v:\tpccback3.dmp'
go
exec sp_addumpdevice 'disk','tpccback4','v:\tpccback4.dmp'
go
```

RESTORE.SQL

```
-- File:    RESTORE.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Loads database backup from backup files


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

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

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

go
```

B.2 Build indices

IDXCUSCL.SQL

```
-- File:    IDXCUSCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates clustered index on customer table


use tpcc
go

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

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

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

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

go
```

IDXCUSNC.SQL

```
-- File:    IDXCUSNC.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates non-clustered index on customer table


use tpcc
go

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

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

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

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

go
```

IDXDISCL.SQL

```
-- File:    IDXDISCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates clustered index on district table


use tpcc
go

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

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

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

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

go
```

IDXITMCL.SQL

```
-- File:    IDXITMCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
```

APPENDIX B – DATABASE DESIGN

```
-- Purpose:  Creates clustered index on item table


use tpcc
go

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

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

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

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

go
```

IDXNODCL.SQL

```
-- File:    IDXNODCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates clustered index on new_order table


use tpcc
go

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

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

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

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

go
```

IDXODLCL.SQL

```
-- File:    IDXODLCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates clustered index on order_line table


use tpcc
go

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

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

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

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

go
```

IDXORDCL.SQL

```
-- File:    IDXORDCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates clustered index on orders table


use tpcc
go

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

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

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

```
                on MSSQL_misc_fg

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

go
```

IDXORDNC.SQL

```
-- File:    IDXORDNC.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates non-clustered index on orders table


use tpcc
go

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

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

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

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

go
```

IDXSTKCL.SQL

```
-- File:    IDXSTKCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates clustered index on stock table

use tpcc
go

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

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

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

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

go
```

IDXWARCL.SQL

```
-- File:    IDXWARCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates clustered index on warehouse table


use tpcc
go

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

if exists ( select name from sysindexes where name = 'warehouse_c1' )
                drop index warehouse.warehouse_c1

create unique clustered index warehouse_c1 on warehouse(w_id)
                with fillfactor=100 on MSSQL_misc_fg

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

go
```

B.3 Database Options

DBOPT1.SQL

```
-- File:    DBOPT1.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Sets database options for data load


use master
go

exec sp_dboption tpcc,'select into/bulkcopy',true
exec sp_dboption tpcc,'trunc. log on chkpt.',true
go

use tpcc
go

checkpoint
go
```

DBOPT2.SQL

```
-- File:    DBOPT2.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Resets database options after data load

sp_dboption tpcc,'select into/bulkcopy',FALSE
GO

sp_dboption tpcc,'trunc. log on chkpt.',FALSE
GO

USE tpcc
GO

CHECKPOINT
GO

sp_configure 'allow updates',1
GO

RECONFIGURE WITH OVERRIDE
GO

DECLARE    @msg                    varchar(50)

--                          --
--      OPTIONS FOR SQL SERVER 8.0      --
-- Set option values for user-defined indexes --
--                          --

SET        @msg        = ' '
PRINT      @msg
SET        @msg        = 'Setting SQL Server indexoptions'
PRINT      @msg
SET        @msg        = ' '
PRINT      @msg

EXEC sp_indexoption      'customer',    'DisAllowPageLocks',      TRUE
EXEC sp_indexoption      'district',    'DisAllowPageLocks',      TRUE
EXEC sp_indexoption      'warehouse',   'DisAllowPageLocks',      TRUE
EXEC sp_indexoption      'stock',       'DisAllowPageLocks',      TRUE
EXEC sp_indexoption      'order_line',  'DisAllowRowLocks',       TRUE
EXEC sp_indexoption      'orders',      'DisAllowRowLocks',       TRUE
EXEC sp_indexoption      'new_order',   'DisAllowRowLocks',       TRUE
EXEC sp_indexoption      'item',                    'DisAllowRowLocks',      TRUE
EXEC sp_indexoption      'item',                    'DisAllowPageLocks',     TRUE
GO

Print ' '
Print '******************'
Print 'Pre-specified Locking Hierarchy:'
Print '          Lockflag = 0 ==> No pre-specified hierarchy'
Print '  Lockflag = 1 ==> Lock at Page-level then Table-level'
Print '  Lockflag = 2 ==> Lock at Row-level then Table-level'
Print '  Lockflag = 3 ==> Lock at Table-level'
Print ' '

SELECT     name,lockflags
FROM       sysindexes
WHERE      object_id('warehouse')      = id OR
           object_id('district')       = id OR
           object_id('customer')       = id OR
           object_id('stock')          = id OR
           object_id('orders')         = id OR
           object_id('order_line')     = id OR
           object_id('history')        = id OR
           object_id('new_order')      = id OR
           object_id('item')           = id
ORDER      BY lockflags asc
GO

sp_configure 'allow updates',0
GO
```

```
RECONFIGURE WITH OVERRIDE
GO

EXEC sp_dboption tpcc,     'auto update statistics',     FALSE
EXEC sp_dboption tpcc,     'auto create statistics',     FALSE
EXEC sp_dboption tpcc,     'torn page detection',                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

tables.SQL

```
-- File:    TABLES.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.21
--          Copyright Microsoft, 1999, 2000
-- Purpose:  Creates TPC-C tables


use tpcc
go

--
-- Remove all existing TPC-C tables
--

if exists ( select name from sysobjects where name = 'warehouse' )
        drop table warehouse
go
if exists ( select name from sysobjects where name = 'district' )
        drop table district
go
if exists ( select name from sysobjects where name = 'customer' )
        drop table customer
go
if exists ( select name from sysobjects where name = 'history' )
        drop table history
go
if exists ( select name from sysobjects where name = 'new_order' )
        drop table new_order
go
if exists ( select name from sysobjects where name = 'orders' )
        drop table orders
go
if exists ( select name from sysobjects where name = 'order_line' )
        drop table order_line
go
if exists ( select name from sysobjects where name = 'item' )
        drop table item
go
if exists ( select name from sysobjects where name = 'stock' )
        drop table stock
go

--
-- Create new tables
--

create table warehouse
(
        w_id                            smallint,
        w_name                          char(10),
        w_street_1              char(20),
        w_street_2              char(20),
        w_city                          char(20),
        w_state                         char(2),
        w_zip                           char(9),
        w_tax                           numeric(4,4),
        w_ytd                           numeric(12,2)
) on MSSQL_misc_fg
go

create table district
(
        d_id                            tinyint,
        d_w_id                          smallint,
        d_name                          char(10),
        d_street_1              char(20),
        d_street_2              char(20),
        d_city                          char(20),
        d_state                         char(2),
        d_zip                           char(9),
        d_tax                           numeric(4,4),
        d_ytd                           numeric(12,2),
        d_next_o_id             int
) on MSSQL_misc_fg
go

create table customer
(
        c_id                            int,
        c_d_id                          tinyint,
        c_w_id                          smallint,
        c_first                         char(16),
        c_middle                char(2),
        c_last                          char(16),
```

```
        c_street_1              char(20),
        c_street_2              char(20),
        c_city                  char(20),
        c_state                 char(2),
        c_zip                   char(9),
        c_phone                 char(16),
        c_since                 datetime,
        c_credit                char(2),
        c_credit_lim    numeric(12,2),
        c_discount              numeric(4,4),
        c_balance               numeric(12,2),
        c_ytd_payment           numeric(12,2),
        c_payment_cnt           smallint,
        c_delivery_cnt          smallint,
        c_data                  char(500)
) on MSSQL_cs_fg
go

create table history
(
        h_c_id                  int,
        h_c_d_id                tinyint,
        h_c_w_id                smallint,
        h_d_id                  tinyint,
        h_w_id                  smallint,
        h_date                  datetime,
        h_amount                numeric(6,2),
        h_data                  char(24)
) on MSSQL_misc_fg
go

create table new_order
(
        no_o_id                 int,
        no_d_id                 tinyint,
        no_w_id                 smallint
) on MSSQL_misc_fg
go

create table orders
(
        o_id                    int,
        o_d_id                  tinyint,
        o_w_id                  smallint,
        o_c_id                  int,
        o_entry_d               datetime,
        o_carrier_id    tinyint,
        o_ol_cnt                tinyint,
        o_all_local             tinyint
) on MSSQL_misc_fg
go

create table order_line
(
        ol_o_id                 int,
        ol_d_id                 tinyint,
        ol_w_id                 smallint,
        ol_number               tinyint,
        ol_i_id                 int,
        ol_supply_w_id          smallint,
        ol_delivery_d   datetime,
        ol_quantity             smallint,
        ol_amount               numeric(6,2),
        ol_dist_info    char(24)
) on MSSQL_misc_fg
go

create table item
(
        i_id                    int,
        i_im_id                 int,
        i_name                  char(24),
        i_price                 numeric(5,2),
        i_data                  char(50)
) on MSSQL_misc_fg
go

create table stock
(
        s_i_id                  int,
        s_w_id                  smallint,
        s_quantity              smallint,
        s_dist_01               char(24),
        s_dist_02               char(24),
        s_dist_03               char(24),
        s_dist_04               char(24),
        s_dist_05               char(24),
        s_dist_06               char(24),
        s_dist_07               char(24),
        s_dist_08               char(24),
        s_dist_09               char(24),
        s_dist_10               char(24),
        s_ytd                   int,
        s_order_cnt             smallint,
        s_remote_cnt            smallint,
        s_data                  char(50)
) on MSSQL_cs_fg
go
```

B.5 Stored Procedures

NEWORD.SQL

APPENDIX B – DATABASE DESIGN

```
-- File:    NEWORD.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates new order transaction stored procedure
--
--                  Interface Level: 4.10.000

use tpcc
go

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. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates delivery transaction stored procedure
--
--              Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_delivery" )
        drop procedure tpcc_delivery
go

create proc tpcc_delivery     @w_id         smallint,
                                                @o_carrier_id   smallint
as

declare @d_id                 tinyint,
    @o_id   int,
    @c_id   int,
    @total  numeric(12,2),
    @oid1   int,
    @oid2   int,
    @oid3   int,
    @oid4   int,
    @oid5   int,
    @oid6   int,
    @oid7   int,
    @oid8   int,
    @oid9   int,
    @oid10  int

select @d_id = 0

begin tran d
```

```
        while (@d_id < 10)
        begin

            select  @d_id  = @d_id + 1,
                    @total = 0,
                    @o_id  = 0

                    select  top 1
                            @o_id       = no_o_id
                    from    new_order (serializable 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. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates order status transaction stored procedure
--
--              Interface Level: 4.10.000

use tpcc
go

if exists ( select name from sysobjects where name = "tpcc_orderstatus" )
            drop procedure   tpcc_orderstatus
go

create proc tpcc_orderstatus        @w_id       smallint,
                                    @d_id       tinyint,

                                    @c_id       int,
                                    @c_last     char(16) = ""

as
```

```
declare @c_balance              numeric(12,2),
        @c_first        char(16),
        @c_middle               char(2),
        @o_id                   int,
        @o_entry_d              datetime,
        @o_carrier_id           smallint,
        @cnt                    smallint

begin tran o

if (@c_id = 0)
        begin

-- get customer id and info using last name

                select  @cnt            = (count(*)+1)/2
                from    customer (repeatableread)
                where   c_last          = @c_last and
                        c_w_id          = @w_id and
                        c_d_id          = @d_id

                set     rowcount @cnt

                select  @c_id                   = c_id,
                        @c_balance      = c_balance,
                          @c_first      = c_first,
                          @c_last       = c_last,
                          @c_middle     = c_middle
                from    customer (repeatableread)
                where   c_last                  = @c_last and
                        c_w_id                  = @w_id and
                        c_d_id                  = @d_id
                order   by c_w_id, c_d_id, c_last, c_first

                set     rowcount 0
        end

        else

        begin

-- get customer info if by id

                select  @c_balance      = c_balance,
                        @c_first        = c_first,
                        @c_middle       = c_middle,
                        @c_last                 = c_last
                from    customer (repeatableread)
                where   c_id                    = @c_id and
                        c_d_id                  = @d_id and
                        c_w_id                  = @w_id

                select  @cnt            = @@rowcount

        end

-- if no such customer

        if (@cnt = 0)
        begin
                raiserror("Customer not found",18,1)
                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
```

PAYMENT.SQL

```
-- File:    PAYMENT.SQL
```

APPENDIX B – DATABASE DESIGN

```
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates payment transaction stored procedure
--
--                  Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_payment" )
                drop procedure tpcc_payment
go

create proc tpcc_payment    @w_id          smallint,
                                                @c_w_id        smallint,
                                                @h_amount      numeric(6,2),
                                                @d_id          tinyint,
                                                @c_d_id        tinyint,
                                                @c_id          int,
                                                @c_last        char(16) = ""


as
declare  @w_street_1    char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   numeric(12,2),
        @c_balance      numeric(12,2),
        @c_discount     numeric(4,4),
        @data           char(500),
        @c_data         char(500),
        @datetime       datetime,
        @w_ytd          numeric(12,2),
        @d_ytd          numeric(12,2),
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
                @d_id_local     tinyint,
                @w_id_local     smallint,
                @c_id_local     int

select @screen_data = ""

begin tran p

-- get payment date

                select          @datetime = getdate()

                if (@c_id = 0)
                begin

-- get customer id and info using last name

                        select          @cnt            = count(*)
                        from            customer (repeatableread)
                        where           c_last          = @c_last and
                                        c_w_id          = @c_w_id and
                                        c_d_id          = @c_d_id


                        select          @val = (@cnt + 1) / 2
                        set             rowcount @val

                        select          @c_id           = c_id
                        from            customer (repeatableread)
                        where           c_last          = @c_last and
                                        c_w_id          = @c_w_id and
                                        c_d_id          = @c_d_id
                        order           by c_last, c_first

                        set             rowcount 0
                end

-- get customer info and update balances

                update          customer
                set             @c_balance      = c_balance     = c_balance - @h_amount,
                                c_payment_cnt                   = c_payment_cnt + 1,
                                c_ytd_payment                   = c_ytd_payment + @h_amount,
                                @c_first        = c_first,
                                @c_middle       = c_middle,
                                @c_last                         = c_last,
                                @c_street_1     = c_street_1,
                                @c_street_2     = c_street_2,
                                @c_city                         = c_city,
                                @c_state        = c_state,
                                @c_zip                          = c_zip,
                                @c_phone        = c_phone,
                                @c_credit       = c_credit,
                                @c_credit_lim                   = c_credit_lim,
```

```
                @c_discount  = c_discount,
                @c_since     = c_since,
                @data                    = c_data,
                @c_id_local  = c_id
        where   c_id                     = @c_id and
                c_w_id                   = @c_w_id and
                c_d_id                   = @c_d_id

-- if customer has bad credit get some more info

        if (@c_credit = "BC")
        begin

--      compute new info

                select @c_data           = convert(char(5),@c_id) +
                                           convert(char(4),@c_d_id) +
                                           convert(char(5),@c_w_id) +
                                           convert(char(4),@d_id) +
                                           convert(char(5),@w_id) +
                                           convert(char(19),@h_amount) +
                                           substring(@data, 1, 458)

-- update customer info

                update  customer
                set     c_data           = @c_data
                where   c_id             = @c_id and
                        c_w_id           = @c_w_id and
                        c_d_id           = @c_d_id

                select  @screen_data = substring (@c_data,1,200)
        end

-- get district data and update year-to-date

        update  district
        set     d_ytd                    = d_ytd + @h_amount,
                @d_street_1  = d_street_1,
                @d_street_2  = d_street_2,
                @d_city                  = d_city,
                @d_state     = d_state,
                @d_zip                   = d_zip,
                @d_name                  = d_name,
                @d_id_local  = d_id
        where   d_w_id                   = @w_id and
                d_id                     = @d_id

-- get warehouse data and update year-to-date

        update  warehouse
        set     w_ytd                    = w_ytd + @h_amount,
                @w_street_1  = w_street_1,
                @w_street_2  = w_street_2,
                @w_city                  = w_city,
                @w_state     = w_state,
                @w_zip                   = w_zip,
                @w_name                  = w_name,
                @w_id_local  = w_id
        where   w_id                     = @w_id

-- create history record

        insert into history values (    @c_id_local,
                                        @c_d_id,
                                        @c_w_id,
                                        @d_id_local,
                                        @w_id_local,
                                        @datetime,
                                        @h_amount,
                                        @w_name + "    " + @d_name)
commit tran p

-- return data to client

select  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data

go
```

STOCKLEV.SQL

```
-- File:    STOCKLEV.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Creates stock level transaction stored procedure
--
--                Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_stocklevel" )
                drop procedure tpcc_stocklevel
go

create proc tpcc_stocklevel                     @w_id           smallint,
                                                @d_id           tinyint,
                                                @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
```

version.sql

```
 -- File:    VERSION.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.22
--          Copyright Microsoft, 2001
-- Purpose:  Returns version level of TPC-C stored procs
-- Note:    Always update the return value of this proc for
--          any interface changes or "must have" bug fixes.
--
-- The value returned by this SP defines the "interface level",
-- which must match between the stored procs and the client code.
-- The interface level may be down rev from the current kit.  This
-- indicates that the interface hasn't changed since that version.

use tpcc
go

if exists ( select name from sysobjects where name = "tpcc_version" )
                drop procedure tpcc_version
go

create proc tpcc_version
as
declare         @version        char(8)

begin
                select @version = "4.10.000"
                select @version as "Version"
end

go
```

B.6 Loader Source Code

TPCC.H

```
//              File:                   TPCC.H
//                                              Microsoft TPC-C Kit Ver. 4.22
//                                              Copyright Microsoft, 1996, 1997, 1998, 1999, 2000, 2001
//              Purpose:        Header file for TPC-C database loader


// Build number of TPC Benchmark Kit
#define TPCKIT_VER      "4.22"

// General headers
#include <windows.h>
#include <winbase.h>
#include <stdlib.h>
#include <stdio.h>
#include <process.h>
#include <stddef.h>
#include <stdarg.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <sys\types.h>

// ODBC headers
```

APPENDIX B – DATABASE DESIGN

```
#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. 4.22
//                                  Copyright Microsoft, 2000, 2001
//          Purpose:    Source file for TPC-C database loader


// Includes
#include "tpcc.h"
#include "search.h"

// Defines
#define MAXITEMS            100000
#define MAXITEMS_SCALE_DOWN                 100
#define CUSTOMERS_PER_DISTRICT  3000
#define CUSTOMERS_SCALE_DOWN    30
#define DISTRICT_PER_WAREHOUSE  10
#define ORDERS_PER_DISTRICT     3000
#define ORDERS_SCALE_DOWN       30
#define MAX_CUSTOMER_THREADS    2
#define MAX_ORDER_THREADS       3
#define MAX_MAIN_THREADS        4

// Functions declarations

void HandleErrorDBC (SQLHDBC  hdbc1);

void CheckSQL();
void CheckDataBase();

long NURand();
void LoadItem();
void LoadWarehouse();

void Stock();
void District();

void LoadCustomer();
void CustomerBufInit();
void CustomerBufLoad();
void LoadCustomerTable();
void LoadHistoryTable();
void LoadOrders();
void OrdersBufInit();
void OrdersBufLoad();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void GetPermutation();
void CheckForCommit();
void OpenConnections();
void BuildIndex();
void FormatDate ();

// Shared memory structures

typedef struct
{
   long            ol;
   long            ol_i_id;
   short           ol_supply_w_id;
   short           ol_quantity;
   double           ol_amount;
   char            ol_dist_info[DIST_INFO_LEN+1];
                   char                                    ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCT;

typedef struct
{
   long            o_id;
   short           o_d_id;
   short           o_w_id;
   long            o_c_id;
   short           o_carrier_id;
   short           o_ol_cnt;
   short           o_all_local;
                   ORDER_LINE_STRUCT   o_ol[15];
} ORDERS_STRUCT;

typedef struct
{
   long                                    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;
```

APPENDIX B – DATABASE DESIGN

```
//  fix to avoid ODBC float to numeric conversion problem.
//  double                                  c_balance;
            char                                    c_balance[6];

   double                                   c_ytd_payment;
            short                                   c_payment_cnt;
            short                                   c_delivery_cnt;
            char                                    c_data[C_DATA_LEN+1];
            double                                  h_amount;
   char                                     h_data[H_DATA_LEN+1];
} CUSTOMER_STRUCT;

typedef struct
{
            char                                    c_last[LAST_NAME_LEN+1];
            char                                    c_first[FIRST_NAME_LEN+1];
   long                                     c_id;
} CUSTOMER_SORT_STRUCT;

typedef struct
{
   long             time_start;
} LOADER_TIME_STRUCT;


// Global variables

char        szLastError[300];

HENV        henv;

HDBC        v_hdbc;                                         // for SQL Server version
verification
HDBC        i_hdbc1;                                        // for ITEM table
HDBC        w_hdbc1;                                        // for WAREHOUSE, DISTRICT, STOCK
HDBC        c_hdbc1;                                        // for CUSTOMER
HDBC        c_hdbc2;                                        // for HISTORY
HDBC        o_hdbc1;                                        // for ORDERS
HDBC        o_hdbc2;                                        // for NEW-ORDER

HDBC        o_hdbc3;                                        // for ORDER-LINE

HSTMT       v_hstmt;                                        // for SQL Server version verification
HSTMT       i_hstmt1;
HSTMT       w_hstmt1;
HSTMT       c_hstmt1, c_hstmt2;
HSTMT       o_hstmt1, o_hstmt2, o_hstmt3;

ORDERS_STRUCT   orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT customer_buf[CUSTOMERS_PER_DISTRICT];
long        orders_rows_loaded;
long        new_order_rows_loaded;
long        order_line_rows_loaded;
long        history_rows_loaded;
long        customer_rows_loaded;
long        stock_rows_loaded;
long        district_rows_loaded;
long        item_rows_loaded;
long        warehouse_rows_loaded;
long        main_time_start;
long        main_time_end;
long                                max_items;
long                                customers_per_district;
long                                orders_per_district;
long                                first_new_order;
long                                last_new_order;


TPCCLDR_ARGS    *aptr, args;


//============================================================================
//
// Function name: main
//
//============================================================================

int main(int  argc, char **argv)
{

            DWORD       dwThreadID[MAX_MAIN_THREADS];
            HANDLE      hThread[MAX_MAIN_THREADS];
            FILE        *fLoader;
            char        buffer[255];
            int                                             i;

            for (i=0; i<MAX_MAIN_THREADS; i++)
                  hThread[i] = NULL;

   printf("\n*************************************************************");
            printf("\n*                                                   *");
            printf("\n*  Microsoft SQL Server                             *");
            printf("\n*                                                   *");
            printf("\n*  TPC-C BENCHMARK KIT:  Database loader            *");
            printf("\n*  Version %s                                       *", TPCKIT_VER);
            printf("\n*                                                   *");
            printf("\n*************************************************************\n\n");

   // process command line arguments

   aptr = &args;
   GetArgsLoader(argc, argv, aptr);

            // verify database and tables exist before attempting to load

            CheckSQL();
            CheckDataBase();

            printf("Build interface is ODBC.\n");
```

```
		if (aptr->build_index == 0)
			printf("Data load only - no index creation.\n");
		else
			printf("Data load and index creation.\n");

		if (aptr->index_order == 0)
			printf("Clustered indexes will be created after bulk load.\n");
		else
			printf("Clustered indexes will be created before bulk load.\n");

		// set database scale values
		if (aptr->scale_down == 1)
		{
			printf("*** Scaled Down Database ***\n");
			max_items = MAXITEMS_SCALE_DOWN;
			customers_per_district = CUSTOMERS_SCALE_DOWN;
			orders_per_district = ORDERS_SCALE_DOWN;
			first_new_order = 0;
			last_new_order  = 30;
		}
		else
		{
			max_items = MAXITEMS;
			customers_per_district = CUSTOMERS_PER_DISTRICT;
			orders_per_district = ORDERS_PER_DISTRICT;
			first_new_order = 2100;
			last_new_order  = 3000;
		}


		// open connections to SQL Server

		OpenConnections();

		// open file for loader results
fLoader = fopen(aptr->loader_res_file, "w");

if (fLoader == NULL)
		{
			printf("Error, loader result file open failed.");
			exit(-1);
		}

// start loading data

sprintf(buffer,"TPC-C load started for %ld warehouses.\n",aptr->num_warehouses);

		printf("%s",buffer);
		fprintf(fLoader,"%s",buffer);

		main_time_start = (TimeNow() / MILLI);

		// start parallel load threads

		if (aptr->tables_all || aptr->table_item)
		{
			fprintf(fLoader, "\nStarting loader threads for: item\n");

			hThread[0] = CreateThread(NULL,
                                                                                0,
(LPTHREAD_START_ROUTINE) LoadItem,
                                                                                NULL,
                                                                                0,
                                                                                &dwThreadID[0]);

			if (hThread[0] == NULL)
			{
				printf("Error, failed in creating creating thread = 0.\n");
				exit(-1);
			}
		}

		if (aptr->tables_all || aptr->table_warehouse)
		{
			fprintf(fLoader, "Starting loader threads for: warehouse\n");

			hThread[1] = CreateThread(NULL,
                                                                                0,
(LPTHREAD_START_ROUTINE) LoadWarehouse,
                                                                                NULL,
                                                                                0,
                                                                                &dwThreadID[1]);

			if (hThread[1] == NULL)
			{
				printf("Error, failed in creating creating thread = 1.\n");
				exit(-1);
			}
		}

		if (aptr->tables_all || aptr->table_customer)
		{
			fprintf(fLoader, "Starting loader threads for: customer\n");

			hThread[2] = CreateThread(NULL,
                                                                                0,
(LPTHREAD_START_ROUTINE) LoadCustomer,
                                                                                NULL,
                                                                                0,
                                                                                &dwThreadID[2]);

			if (hThread[2] == NULL)
			{
				printf("Error, failed in creating creating main thread = 2.\n");
				exit(-1);
			}
```

```
		}

		if (aptr->tables_all || aptr->table_orders)
		{
			fprintf(fLoader, "Starting loader threads for: orders\n");

			hThread[3] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadOrders,
NULL,
0,
&dwThreadID[3]);

			if (hThread[3] == NULL)
			{
				printf("Error, failed in creating creating main thread = 3.\n");
				exit(-1);
			}
		}

		// Wait for threads to finish...
		for (i=0; i<MAX_MAIN_THREADS; i++)
		{
			if (hThread[i] != NULL)
			{
				WaitForSingleObject( hThread[i], INFINITE );
				CloseHandle(hThread[i]);
				hThread[i] = NULL;
			}
		}

		main_time_end = (TimeNow() / MILLI);

	sprintf(buffer,"\nTPC-C load completed successfully in %ld minutes.\n",
				(main_time_end - main_time_start)/60);

		printf("%s",buffer);
		fprintf(fLoader, "%s", buffer);

		fclose(fLoader);

		SQLFreeEnv(henv);

	exit(0);

		return 0;
}


//==============================================================================
//
// Function name: LoadItem
//
//==============================================================================

void LoadItem()
{
	long		i_id;
		long		i_im_id;
	char		i_name[I_NAME_LEN+1];
	double		i_price;
	char		i_data[I_DATA_LEN+1];
		char		name[20];
		long		time_start;
		RETCODE		rc;
		DBINT		rcint;
		char		bcphint[128];

		// Seed with unique number
		seed(1);

		printf("Loading item table...\n");

		// if build index before load
		if ((aptr->build_index == 1) && (aptr->index_order == 1))
			BuildIndex("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,
&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,
&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,
0,
					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))
                {
```

APPENDIX B – DATABASE DESIGN

```
                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;
        }

}
```

APPENDIX B – DATABASE DESIGN

```
//====================================================================
//
// 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   : CheckSQL
//
//========================================================================

void CheckSQL()
{

        RETCODE       rc;

        char                            szDriverStr ing[300];
        char                            szDriverStringOut[1024];
        int                                     SQLBuildFlag;
        char                            resp;

        SQLSMALLINT                     cbDriverStringOut;
        SQLCHAR                         SQLVersion[19];
        SQLINTEGER                      SQLVersionInd;



        SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv );

        SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0 );

        SQLAllocHandle(SQL_HANDLE_DBC, henv , &v_hdbc);

        SQLSetConnectAttr(v_hdbc, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );

        // Open connection to SQL Server

        sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s" ,
                                                                        aptr->server,
                                                                        aptr->user,
                                                                        aptr->password );

        if ( SQLSetConnectAttr( v_hdbc, SQL_ATTR_PACKET_SIZE, (SQLPOINTER)aptr->pack_size,
SQL_IS_UINTEGER ) != SQL_SUCCESS )
                HandleErrorDBC(v_hdbc);

        rc = SQLDriverConnect ( v_hdbc,
                                                                NULL,
                                                                (SQLCHAR*)&szDriverString[0] ,
                                                                SQL_NTS,
                                                                (SQLCHAR*)&szDriverStringOut[0],
                                                                sizeof(szDriverStringOut),
                                                                &cbDriverStringOut,
                                                                SQL_DRIVER_NOPROMPT );

        if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
                HandleErrorDBC(v_hdbc);

        if ( SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt) != SQL_SUCCESS )
                HandleErrorSTMT(v_hstmt);

        rc = SQLBindCol(v_hstmt, 4, SQL_C_CHAR, &SQLVersion, sizeof(SQLVersion), &SQLVersionInd);

        // issue SQL Server extended stored procedure (xp_msver) to determine installed version
        rc = SQLExecDirect(v_hstmt, "EXECUTE xp_msver ProductVersion", SQL_NTS);

        if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
                HandleErrorSTMT(v_hstmt);

        rc = SQLFetch(v_hstmt);

        if (rc != SQL_SUCCESS)
                HandleErrorDBC(v_hdbc);

        // Check build number to ensure 8.00.194 or higher
```

APPENDIX B – DATABASE DESIGN

```
        SQLBuildFlag = 1;

        // first check the Major version

        if ( SQLVersion[0] == '8' )
        {
                if (( SQLVersion[2] == '0' ) & ( SQLVersion[3] == '0' )      )
                {
                        if ( SQLVersion[5] == '1' )
                        {
                                if ( (SQLVersion[6] == '9') &  (SQLVersion[7] == '4') )
                                {
                                        SQLBuildFlag = 0;
                                        printf("You are using SQL Server version = %9s\n\n",
SQLVersion);
                                }
                                else
                                {
                                        SQLBuildFlag = 1;
                                }
                        }
                        else
                        {
                                if ( SQLVersion[5] == '3' )
                                {
                                        if ( (SQLVersion[6] >= 53) & (SQLVersion[7] >= 48) )
                                        {
                                                SQLBuildFlag = 0;
                                                printf("You are using SQL Server version =
%9s\n\n", SQLVersion);
                                        }
                                        else
                                        {
                                                SQLBuildFlag = 1;
                                        }
                                }
                        }
                }
        }
        else
        {
                SQLBuildFlag = 1;
        }

        if ( SQLBuildFlag == 1 )
        {
                printf("NOTE:  The SQL Server version you are using is not supported\n");
                printf("for TPC-C benchmarking.  You currently have SQL Server version %9s\n",SQLVersion);
                printf("installed.  Please upgrade to Microsoft SQL Server 2000 (8.00.0194) or better.\n");
                printf("and re-run the SETUP program.\n\n");
                printf("Do you wish to continue with setup? (Y/N): ");
                resp = getchar();
                if ( ( resp == 'N' ) || (resp == 'n') )
                {
                        printf("\nSetup Aborted!\n");
                        exit(1);
                }
        }

        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
        SQLDisconnect(v_hdbc);
        SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

        return;

}


//============================================================================
//
// Function   : CheckDataBase
//
//============================================================================

void CheckDataBase()
{

        RETCODE       rc;

        char                    szDriverString[300];
        char                    szDriverStringOut[1024];
        char                    TablesBitMap[9] = {"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. 4.22
//			Copyright Microsoft, 1996, 1997, 1998, 1999, 2000, 2001
//	Purpose:	Source file for command line processing


// Includes
#include "tpcc.h"

//===========================================================================
//
// Function name: GetArgsLoader
//
//===========================================================================

void GetArgsLoader(int argc, char **argv, TPCCLDR_ARGS *pargs)
{
   int		i;
   char		*ptr;

#ifdef DEBUG
   printf("[%ld]DBG: Entering GetArgsLoader()\n", (int) GetCurrentThreadId());
#endif
```

APPENDIX B – DATABASE DESIGN

```
/* 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. 4.22
//                                                      Copyright Microsoft, 1996, 1997, 1998, 1999, 2000, 2001
//              Purpose:        Random number generation routines for database loader


// Includes
#include "tpcc.h"
#include "math.h"

// Defines
#define A          16807
#define M       2147483647
#define Q          127773      /* M div A */
#define R           2836       /* M mod A */
#define Thread                  __declspec(thread)

// Globals
long            Thread Seed = 0;      /* thread local seed */

/*****************************************************************************
*                                                                            *
* random -                                                                   *
*      Implements a GOOD pseudo random number generator.  This generator     *
*      will/should? run the complete period before repeating.                *
*                                                                            *
* Copied from:                                                               *
*      Random Numbers Generators: Good Ones Are Hard to Find.                *
*      Communications of the ACM - October 1988 Volume 31 Number 10          *
*                                                                            *
* Machine Dependencies:                                                      *
*      long must be 2 ^ 31 - 1 or greater.                                   *
*                                                                            *
*****************************************************************************/
```

APPENDIX B – DATABASE DESIGN

```
/************************************************************************
* 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. 4.22
//                                              Copyright Microsoft, 1996, 1997, 1998, 1999, 2000, 2001
//              Purpose:        Source file for database loader string functions


// Includes
#include "tpcc.h"
#include <string.h>
#include <ctype.h>

//==============================================================================
//
// Function name: MakeAddress
//
//==============================================================================

void MakeAddress(char *street_1,
                                                char *street_2,
                                                char *city,
                                                char *state,
                                                char *zip)
{

#ifdef DEBUG
   printf("[%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;

}


//==============================================================================
//
```

APPENDIX B – DATABASE DESIGN

```
// 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;
}
```

TIME.C

```
//              File:                   TIME.C
//                                              Microsoft TPC-C Kit Ver. 4.22
//                                              Copyright Microsoft, 1996, 1997, 1998, 1999, 2000, 2001
//              Purpose:        Source file for time functions


// Includes
#include "tpcc.h"

// Globals
static long start_sec;


//==============================================================================
//
// Function name: TimeNow
//
//==============================================================================

long TimeNow()
{
  long                          time_now;
  struct        _timeb el_time;

#ifdef DEBUG
   printf("[%ld]DBG: Entering TimeNow()\n", (int) GetCurrentThreadId());
#endif

  _ftime(&el_time);

  time_now = ((el_time.time - start_sec) * 1000) + el_time.millitm;

  return time_now;
}
```

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
?? Distributed File System
?? Distributed Link Tracking Client
?? DNS Client
?? FAX Service
?? File Replication
?? FTP Publishing Service
?? Indexing Service
?? IPSEC Policy Agent
?? License Logging Service
?? Messenger
?? Microsoft Search
?? Mylex Global Array Manager Server
?? Print Spooler
?? Process Control Service
?? Remote Registry Service
?? Run as Service
?? Task scheduler
?? Telnet
?? 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 LH 6000

System Information report written at: 08/08/2001 09:55:22 AM
[System Information]

[Following are sub-categories of this main category]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Advanced Server
Version	5.0.2195 Service Pack 2 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	PRF_SUT5
System Manufacturer	Hewlett Packard
System Model	HP NetServer
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
BIOS Version	05/17/01
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition1
Locale	United States
User Name	PRF_SUT5\Administrator
Time Zone	Pacific Daylight Time
Total Physical Memory	8,183,072 KB
Available Physical Memory	7,951,672 KB
Total Virtual Memory	17,248,672 KB
Available Virtual Memory	16,921,172 KB
Page File Space	9,065,600 KB
Page File	C:\pagefile.sys

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

Resource	Device
IRQ 3	Communications Port (COM2)
IRQ 3	HP NetServer SMIC Controller

[DMA]

Channel	Device	Status
4	Direct memory access controller	OK
3	ECP Printer Port (LPT1)	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
0x0D00-0x0FFF	PCI bus	OK
0x1400-0x184F	PCI bus	OK
0x1400-0x184F	ATI Technologies Inc. 3D RAGE IIC PCI	OK
0x4000-0xFFFF	PCI bus	OK
0x4000-0xFFFF	PCI bus	OK
0x4000-0xFFFF	HP NetServer PCI Hot-Plug Controller	OK
0x1800-0x183F	HP NetServer 10/100TX PCI LAN Adapter #2	OK
0x03B0-0x03BB	ATI Technologies Inc. 3D RAGE IIC PCI	OK
0x03C0-0x03DF	ATI Technologies Inc. 3D RAGE IIC PCI	OK
0x0A79-0x0A79	ISAPNP Read Data Port	OK
0x0279-0x0279	ISAPNP Read Data Port	OK
0x02F4-0x02F7	ISAPNP Read Data Port	OK
0x0081-0x008F	Direct memory access controller	OK
0x00C0-0x00DF	Direct memory access controller	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
0x1100-0x111F	Motherboard resources	OK
0x1200-0x121F	Motherboard resources	OK
0x00E8-0x00E9	Motherboard resources	OK
0x0C00-0x0C01	Motherboard resources	OK
0x0C0B-0x0C0B	Motherboard resources	OK
0x0C14-0x0C14	Motherboard resources	OK
0x0C20-0x0C27	Motherboard resources	OK
0x0C49-0x0C4A	Motherboard resources	OK
0x0C50-0x0C52	Motherboard resources	OK
0x0C6C-0x0C6C	Motherboard resources	OK
0x0C6F-0x0C6F	Motherboard resources	OK
0x0CD6-0x0CD7	Motherboard resources	OK
0x0F50-0x0F58	Motherboard resources	OK
0x0060-0x0060	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard	OK
0x0064-0x0064	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard	OK

APPENDIX C – TUNABLE PARAMETERS

0x0070-0x0071 System CMOS/real time clock OK
0x03F8-0x03FF Communications Port (COM1) OK
0x02F8-0x02FF Communications Port (COM2) OK
0x0378-0x037F ECP Printer Port (LPT1) OK
0x0778-0x077F ECP Printer Port (LPT1) OK
0x03F0-0x03F5 Standard floppy disk controller OK
0x03F7-0x03F7 Standard floppy disk controller OK
0x1840-0x184F Standard Dual Channel PCI IDE Controller OK
0x01F0-0x01F7 Primary IDE Channel OK
0x03F6-0x03F6 Primary IDE Channel OK
0x0170-0x0177 Secondary IDE Channel OK
0x0376-0x0376 Secondary IDE Channel OK
0x2000-0x3FFF PCI bus OK
0x2000-0x3FFF DEC 21154 PCI to PCI bridge OK
0x2000-0x3FFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0x3000-0x3FFF DEC 21154 PCI to PCI bridge OK
0x3000-0x3FFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0x4400-0x44FF HP NetServer SMIC Controller OK
0x5000-0x5FFF PCI standard PCI-to-PCI bridge OK
0x5000-0x5FFF Symbios Logic 896, 22910 PCI SCSI Adapter OK
0x5400-0x54FF Symbios Logic 896, 22910 PCI SCSI Adapter OK
0x6000-0x6FFF DEC 21154 PCI to PCI bridge OK
0x6000-0x6FFF Mylex eXtremeRAID 2000 Disk Array Controller OK
0x7000-0x7FFF DEC 21154 PCI to PCI bridge OK
0x7000-0x7FFF Mylex eXtremeRAID 2000 Disk Array Controller OK

[IRQs]

IRQ Number	Device
9	Microsoft ACPI-Compliant System
18	HP NetServer 10/100TX PCI LAN Adapter #2
13	Numeric data processor
1	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
12	PS/2 Compatible Mouse
8	System CMOS/real time clock
4	Communications Port (COM1)
3	Communications Port (COM2)
3	HP NetServer SMIC Controller
6	Standard floppy disk controller
14	Primary IDE Channel
20	Mylex eXtremeRAID 2000 Disk Array Controller
21	Mylex eXtremeRAID 2000 Disk Array Controller
11	HP NetServer PCI Hot-Plug Controller
27	Symbios Logic 896, 22910 PCI SCSI Adapter
26	Symbios Logic 896, 22910 PCI SCSI Adapter
5	PCI Memory Controller
23	cLAN Host Adapter
25	Mylex eXtremeRAID 2000 Disk Array Controller
24	Mylex eXtremeRAID 2000 Disk Array Controller

[Memory]

Range	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	ATI Technologies Inc. 3D RAGE IIC PCI	OK
0xC0000-0xC3FFF	PCI bus	OK
0xC4000-0xC7FFF	PCI bus	OK
0xC8000-0xCBFFF	PCI bus	OK
0xCC000-0xCFFFF	PCI bus	OK
0xD0000-0xD3FFF	PCI bus	OK
0xD4000-0xD7FFF	PCI bus	OK
0xD8000-0xDBFFF	PCI bus	OK
0xDC000-0xDFFFF	PCI bus	OK
0xF3800000-0xF4FFFFFF	PCI bus	OK
0xFE000000-0xFEBEFFFF	PCI bus	OK
0xF3801000-0xF3801FFF	HP NetServer 10/100TX PCI LAN Adapter #2	OK
0xF3900000-0xF39FFFFF	HP NetServer 10/100TX PCI LAN Adapter #2	OK
0xF4000000-0xF4FFFFFF	ATI Technologies Inc. 3D RAGE IIC PCI	OK
0xF3802000-0xF3802FFF	ATI Technologies Inc. 3D RAGE IIC PCI	OK
0xFFFD0000-0xFFFEFFFF	System board	OK
0xF5000000-0xF5FFFFFF	PCI bus	OK
0xF5000000-0xF5FFFFFF	DEC 21154 PCI to PCI bridge	OK
0xF5000000-0xF5FFFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xF6000000-0xF6FFFFFF	PCI bus	OK
0xF6000000-0xF6FFFFFF	DEC 21154 PCI to PCI bridge	OK
0xF6000000-0xF6FFFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xF5800000-0xF5FFFFFF	DEC 21154 PCI to PCI bridge	OK
0xF5800000-0xF5FFFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xF6800000-0xF6FFFFFF	DEC 21154 PCI to PCI bridge	OK
0xF6800000-0xF6FFFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xF7000000-0xFA7FFFFF	PCI bus	OK
0xF7000000-0xFA7FFFFF	cLAN Host Adapter	OK
0xFA800000-0xFCFFFFFF	PCI bus	OK
0xF9000000-0xF90FFFFF	PCI standard PCI-to-PCI bridge	OK
0xF9000000-0xF90FFFFF	HP NetServer Smart IRQ Router	OK
0xF900C000-0xF900C3FF	Symbios Logic 896, 22910 PCI SCSI Adapter	OK
0xF9008000-0xF9009FFF	Symbios Logic 896, 22910 PCI SCSI Adapter	OK
0xF900C400-0xF900C7FF	Symbios Logic 896, 22910 PCI SCSI Adapter	OK
0xF900A000-0xF900BFFF	Symbios Logic 896, 22910 PCI SCSI Adapter	OK
0xFB000000-0xFBFFFFFF	PCI Memory Controller	OK
0xF7020000-0xF703FFFF	cLAN Host Adapter	OK
0xF7200000-0xF73FFFFF	cLAN Host Adapter	OK
0xF8000000-0xF8FFFFFF	cLAN Host Adapter	OK
0xF9800000-0xF9FFFFFF	DEC 21154 PCI to PCI bridge	OK
0xF9800000-0xF9FFFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xFC000000-0xFC7FFFFF	DEC 21154 PCI to PCI bridge	OK
0xFC000000-0xFC7FFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xFA000000-0xFA7FFFFF	DEC 21154 PCI to PCI bridge	OK
0xFA000000-0xFA7FFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK
0xFC800000-0xFCFFFFFF	DEC 21154 PCI to PCI bridge	OK
0xFC800000-0xFCFFFFFF	Mylex eXtremeRAID 2000 Disk Array Controller	OK

[Components]

[Following are sub-categories of this main category]

[Multimedia]

[Following are sub-categories of this main category]

[Audio Codecs]

APPENDIX C – TUNABLE PARAMETERS

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\msg723.acm Microsoft Corporation OK
C:\WINNT\System32\MSG723.ACM 4.4.3385 106.77 KB (109,328 bytes) 5/4/2001 9:41:18 AM
c:\winnt\system32\lhacm.acm Microsoft Corporation OK
C:\WINNT\System32\LHACM.ACM 4.4.3385 33.27 KB (34,064 bytes) 5/4/2001 9:41:19 AM
c:\winnt\system32\tssoft32.acm DSP GROUP, INC. OK
C:\WINNT\System32\TSSOFT32.ACM 1.01 9.27 KB (9,488 bytes) 12/7/1999 4:00:00 AM
c:\winnt\system32\msg711.acm Microsoft Corporation OK
C:\WINNT\System32\MSG711.ACM 5.00.2134.1 10.27 KB (10,512 bytes) 12/7/1999 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\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
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

[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) 5/4/2001 9:41:18 AM
c:\winnt\system32\msh263.drv Microsoft Corporation OK
C:\WINNT\System32\MSH263.DRV 4.4.3385 252.27 KB (258,320 bytes) 5/4/2001 9:40:31 AM
c:\winnt\system32\msvidc32.dll Microsoft Corporation OK
C:\WINNT\System32\MSVIDC32.DLL 5.00.2134.1 27.27 KB (27,920 bytes) 12/7/1999 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 HITACHI CDR-8330
Manufacturer (Standard CD-ROM drives)
Status OK
Transfer Rate Not Available
SCSI Target ID 0
PNP Device ID IDE\CDROMHITACHI_CDR-8330_____0007____\5&A1AB53B&0&0.0.0

[Sound Device]

Item Value
No sound devices

[Display]

Item Value
Name ATI Technologies Inc. 3D RAGE IIC PCI
PNP Device ID PCI\VEN_1002&DEV_4759&SUBSYS_00000000&REV_7A\3&291BF6FF&0&38
Adapter Type ATI 3D RAGE IIC PCI (A21), ATI Technologies Inc. compatible
Adapter Description ATI Technologies Inc. 3D RAGE IIC PCI
Adapter RAM 2.00 MB (2,097,152 bytes)
Installed Drivers atiraged.dll
Driver Version 5.00.2174.1
INF File display.inf (atirage section)
Color Planes 1
Color Table Entries 256
Resolution 1024 x 768 x 75 hertz
Bits/Pixel 8

[Infrared]

Item Value
No infrared devices

[Input]

[Following are sub-categories of this main category]

[Keyboard]

Item Value
Description Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
Name Enhanced (101- or 102-key)
Layout 00000409
PNP Device ID ACPI\PNP0303\5&2C631CF0&0
NumberOfFunctionKeys 12

[Pointing Device]

Item Value
Hardware Type PS/2 Compatible Mouse
Number of Buttons 3
Status OK
PNP Device ID ACPI\PNP0F13\5&2C631CF0&0
Power Management Supported False
Double Click Threshold 6
Handedness Right Handed Operation

[Modem]

Item Value
No modems

[Network]

APPENDIX C – TUNABLE PARAMETERS

[Following are sub-categories of this main category]

[Adapter]

Item Value
Name [00000000] HP NetServer 10/100TX PCI LAN Adapter
Adapter Type Not Available
Product Name HP NetServer 10/100TX PCI LAN Adapter
Installed True
PNP Device ID Not Available
Last Reset 8/8/2001 2:38:36 AM
Index 0
Service Name HPTX
IP Address 16.75.206.169
IP Subnet 255.255.248.0
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:FA:00:0E:9F
Service Name Not Available

Name [00000001] RAS Async Adapter
Adapter Type Not Available
Product Name RAS Async Adapter
Installed True
PNP Device ID Not Available
Last Reset 8/8/2001 2:38:36 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 8/8/2001 2:38:36 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 8/8/2001 2:38:36 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 8/8/2001 2:38:36 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 8/8/2001 2:38:36 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

APPENDIX C – TUNABLE PARAMETERS

DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name NdisWan
Driver c:\winnt\system32\drivers\ndiswan.sys (90096, 5.00.2195.2779)

Name [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 8/8/2001 2:38:36 AM
Index 6
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name NdisWan
Driver c:\winnt\system32\drivers\ndiswan.sys (90096, 5.00.2195.2779)

Name [00000007] 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&291BF6FF&0&30
Last Reset 8/8/2001 2:38:36 AM
Index 7
Service Name HPTX
IP Address 15.75.205.181
IP Subnet 255.255.248.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:10:83:FA:03:CB
Service Name HPTX
IRQ Number 18
I/O Port 0x1800-0x183F
Driver c:\winnt\system32\drivers\hptxnt5.sys (85264, 4.02.27.0000)

Name [00000008] HP NetServer 10/100TX PCI LAN Adapter
Adapter Type Not Available
Product Name HP NetServer 10/100TX PCI LAN Adapter
Installed True
PNP Device ID Not Available
Last Reset 8/8/2001 2:38:36 AM
Index 8
Service Name HPTX
IP Address 16.75.206.169
IP Subnet 255.255.248.0
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:FA:00:0E:9F
Service Name Not Available

Name [00000009] HP NetServer 10/100TX PCI LAN Adapter
Adapter Type Not Available
Product Name HP NetServer 10/100TX PCI LAN Adapter
Installed True
PNP Device ID Not Available
Last Reset 8/8/2001 2:38:36 AM
Index 9
Service Name HPTX
IP Address 16.75.206.169
IP Subnet 255.255.248.0
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:FA:00:0E:9F
Service Name Not Available

Name [00000010] cLAN Host Adapter
Adapter Type Ethernet 802.3
Product Name cLAN Host Adapter
Installed True
PNP Device ID PCI\VEN_135B&DEV_0001&SUBSYS_00000000&REV_00\3&12F48E42&0&20
Last Reset 8/8/2001 2:38:36 AM
Index 10
Service Name GNINDIS
IP Address 16.75.206.169
IP Subnet 255.255.248.0
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:FA:00:0E:9F
Service Name GNINDIS
IRQ Number 23
Driver c:\winnt\system32\drivers\gnindis.sys (22726, 4.2.0.75)

[Protocol]

Item Value
Name MSAFD Tcpip [TCP/IP]
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 16 bytes
MaximumMessageSize 0 bytes
MessageOriented False

APPENDIX C – TUNABLE PARAMETERS

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_{DC2B4961-250A-47A6-BBA5-407B1429D35C}] SEQPACKET 6
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{DC2B4961-250A-47A6-BBA5-407B1429D35C}] DATAGRAM 6
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{672F206B-25E7-4546-9535-913C0B593E6E}] SEQPACKET 5
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False

SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{672F206B-25E7-4546-9535-913C0B593E6E}] DATAGRAM 5
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{0C9D9BF4-BA9B-45EA-A55B-CFEE7C574424}] SEQPACKET 4
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{0C9D9BF4-BA9B-45EA-A55B-CFEE7C574424}] DATAGRAM 4
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{B17B0E16-0353-45D0-A718-1870CC87555A}] SEQPACKET 3
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{B17B0E16-0353-45D0-A718-1870CC87555A}] DATAGRAM 3
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{99ADC03D-F14E-4EC6-AF4F-B0CF514DDBDF}] 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

APPENDIX C – TUNABLE PARAMETERS

SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{99ADC03D-F14E-4EC6-AF4F-B0CF514DDBDF}] 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_{C3ABA2CE-ED6D-443D-849E-03B140CE7B1B}] 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_{C3ABA2CE-ED6D-443D-849E-03B140CE7B1B}] 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_{0E024829-CE9E-4802-A3FB-276AF20EC010}] 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_{0E024829-CE9E-4802-A3FB-276AF20EC010}] DATAGRAM 2
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

[WinSock]

Item Value
File c:\winnt\system32\winsock.dll
Version 3.10
Size 2.80 KB (2,864 bytes)

File c:\winnt\system32\wsock32.dll
Version 5.00.2195.2871
Size 21.27 KB (21,776 bytes)

[Ports]

[Following are sub-categories of this main category]

[Serial]

APPENDIX C – TUNABLE PARAMETERS

Item	Value
Name	COM1
Status	OK
PNP Device ID	ACPI\PNP0501\1
Maximum Input Buffer Size	0
Maximum Output Buffer Size	False
Settable Baud Rate	True
Settable Data Bits	True
Settable Flow Control	True
Settable Parity	True
Settable Parity Check	True
Settable Stop Bits	True
Settable RLSD	True
Supports RLSD	True
Supports 16 Bit Mode	False
Supports Special Characters	False
Baud Rate	9600
Bits/Byte	8
Stop Bits	1
Parity	None
Busy	0
Abort Read/Write on Error	0
Binary Mode Enabled	-1
Continue XMit on XOff	0
CTS Outflow Control	0
Discard NULL Bytes	0
DSR Outflow Control	0
DSR Sensitivity	0
DTR Flow Control Type	Enable
EOF Character	0
Error Replace Character	0
Error Replacement Enabled	0
Event Character	0
Parity Check Enabled	0
RTS Flow Control Type	Enable
XOff Character	19
XOffXMit Threshold	512
XOn Character	17
XOnXMit Threshold	2048
XOnXOff InFlow Control	0
XOnXOff OutFlow Control	0
IRQ Number	4
I/O Port	0x03F8-0x03FF
Driver	c:\winnt\system32\drivers\serial.sys (62416, 5.00.2195.2780)

Name	COM2
Status	OK
PNP Device ID	ACPI\PNP0501\2
Maximum Input Buffer Size	0
Maximum Output Buffer Size	False
Settable Baud Rate	True
Settable Data Bits	True
Settable Flow Control	True
Settable Parity	True
Settable Parity Check	True
Settable Stop Bits	True
Settable RLSD	True
Supports RLSD	True
Supports 16 Bit Mode	False
Supports Special Characters	False
Baud Rate	9600
Bits/Byte	8
Stop Bits	1
Parity	None
Busy	0
Abort Read/Write on Error	0
Binary Mode Enabled	-1
Continue XMit on XOff	0
CTS Outflow Control	0
Discard NULL Bytes	0
DSR Outflow Control	0
DSR Sensitivity	0
DTR Flow Control Type	Enable
EOF Character	0
Error Replace Character	0
Error Replacement Enabled	0
Event Character	0
Parity Check Enabled	0
RTS Flow Control Type	Enable
XOff Character	19
XOffXMit Threshold	512
XOn Character	17
XOnXMit Threshold	2048
XOnXOff InFlow Control	0
XOnXOff OutFlow Control	0
IRQ Number	3
I/O Port	0x02F8-0x02FF
Driver	c:\winnt\system32\drivers\serial.sys (62416, 5.00.2195.2780)

[Parallel]

Item	Value
Name	LPT1
PNP Device ID	ACPI\PNP0401\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)

APPENDIX C – TUNABLE PARAMETERS

Free Space 4.48 GB (4,813,688,832 bytes)
Volume Name
Volume Serial Number 1CD5D6C4
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 80-1208 SCSI Disk Device
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 1
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 3
Drive SCSITargetId 2
Drive SectorsPerTrack 63
Drive Size 9097159680 bytes
Drive TotalCylinders 1106
Drive TotalSectors 17767890
Drive TotalTracks 282030
Drive TracksPerCylinder 255

Drive E:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #1, Partition #0
Partition Size 101.62 GB (109,116,532,224 bytes)
Starting Offset 32256 bytes
Drive Description \\.\PHYSICALDRIVE1
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 1
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 4
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 109116564480 bytes
Drive TotalCylinders 13266
Drive TotalSectors 213118290
Drive TotalTracks 3382830
Drive TracksPerCylinder 255

Drive F:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #2, Partition #0
Partition Size 30.76 GB (33,032,692,224 bytes)
Starting Offset 32256 bytes
Drive Description \\.\PHYSICALDRIVE2
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 3
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 5
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 872948966400 bytes
Drive TotalCylinders 106130
Drive TotalSectors 1704978450
Drive TotalTracks 27063150
Drive TracksPerCylinder 255

Drive G:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #3, Partition #0
Partition Size 30.76 GB (33,032,692,224 bytes)
Starting Offset 32256 bytes
Drive Description \\.\PHYSICALDRIVE3
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 3
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 6
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 872948966400 bytes
Drive TotalCylinders 106130
Drive TotalSectors 1704978450
Drive TotalTracks 27063150
Drive TracksPerCylinder 255

Drive H:

APPENDIX C – TUNABLE PARAMETERS

Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #4, Partition #0
Partition Size 30.76 GB (33,032,692,224 bytes)
Starting Offset 32256 bytes
Drive Description \\.\PHYSICALDRIVE4
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 7
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 872948966400 bytes
Drive TotalCylinders 106130
Drive TotalSectors 1704978450
Drive TotalTracks 27063150
Drive TracksPerCylinder 255

Drive M:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #2, Partition #1
Partition Size 69.33 GB (74,447,009,280 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALDRIVE2
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 3
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 5
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 872948966400 bytes
Drive TotalCylinders 106130
Drive TotalSectors 1704978450
Drive TotalTracks 27063150
Drive TracksPerCylinder 255

Drive N:
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 69.33 GB (74,447,009,280 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALDRIVE3
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 3
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 6
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 872948966400 bytes
Drive TotalCylinders 106130
Drive TotalSectors 1704978450
Drive TotalTracks 27063150
Drive TracksPerCylinder 255

Drive O:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
Partition Disk #4, Partition #1
Partition Size 69.33 GB (74,447,009,280 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALDRIVE4
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 2
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 7
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 872948966400 bytes
Drive TotalCylinders 106130
Drive TotalSectors 1704978450

APPENDIX C – TUNABLE PARAMETERS

Drive TotalTracks 27063150
Drive TracksPerCylinder 255

Drive V:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 712.90 GB (765,469,229,056 bytes)
Free Space 489.26 GB (525,334,695,936 bytes)
Volume Name New Volume
Volume Serial Number D0DA6AB7
Partition Disk #2, Partition #2
Partition Size 712.90 GB (765,469,232,640 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALDRIVE2
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 3
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 5
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 872948966400 bytes
Drive TotalCylinders 106130
Drive TotalSectors 1704978450
Drive TotalTracks 27063150
Drive TracksPerCylinder 255

Drive W:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 712.90 GB (765,469,229,056 bytes)
Free Space 447.58 GB (480,580,984,832 bytes)
Volume Name New Volume
Volume Serial Number 3C70C07C
Partition Disk #3, Partition #2
Partition Size 712.90 GB (765,469,232,640 bytes)
Starting Offset Not Available
Drive Description \\.\PHYSICALD RIVE3
Drive Manufacturer Not Available
Drive Model Not Available
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 3
Drive SCSIBus 4
Drive SCSILogicalUnit 0
Drive SCSIPort 6
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 872948966400 bytes
Drive TotalCylinders 106130
Drive TotalSectors 1704978450
Drive TotalTracks 27063150
Drive TracksPerCylinder 255

[SCSI]

Item Value
Name Mylex eXtremeRAID 2000 Disk Array Controller
Caption Mylex eXtremeR AID 2000 Disk Array Controller
Driver dac2w2k
Status OK
PNP Device ID PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&1CF81133&0&4010
Device ID PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&1CF81133&0&4010
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 20
I/O Port 0x2000-0x3FFF
Driver c:\winnt\system32\drivers\dac2w2k.sys (185584, 9.00-04)

Name Mylex eXtremeRAID 2000 Disk Array Controller
Caption Mylex eXtremeRAID 2000 Disk Array Controller
Driver dac2w2k
Status OK
PNP Device ID PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&119F9F9E&0&4018
Device ID PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&119F9F9E&0&4018
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 21
I/O Port 0x3000-0x3FFF
Driver c:\winnt\system32\drivers\dac2w2k.sys (185584, 9.00-04)

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_60E0103C&REV_06\4&229D1194&0&2818
Device ID PCI\VEN_1000&DEV_000B&SUBSYS_60E0103C&REV_06\4&229D1194&0&2818
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 27
I/O Port 0x5000-0x5FFF
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_60E0103C&REV_06\4&229D1194&0&2918
Device ID PCI\VEN_1000&DEV_000B&SUBSYS_60E0103C&REV_06\4&229D1194&0&2918
Device Map Not Available
Index Not Available

APPENDIX C – TUNABLE PARAMETERS

Max Number Controlled Not Available
IRQ Number 26
I/O Port 0x5400-0x54FF
Driver c:\winnt\system32\drivers\sym_hi.sys (21136, 5.00.2134.1)

Name Mylex eXtremeRAID 2000 Disk Array Controller
Caption Mylex eXtremeRAID 2000 Disk Array Controller
Driver dac2w2k
Status OK
PNP Device ID PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&1ECE4987&0&4028
Device ID PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&1ECE4987&0&4028
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 25
I/O Port 0x6000-0x6FFF
Driver c:\winnt\system32\drivers\dac2w2k.sys (185584, 9.00-04)

Name Mylex eXtremeRAID 2000 Disk Array Controller
Caption Mylex eXtremeRAID 2000 Disk Array Controller
Driver dac2w2k
Status OK
PNP Device ID PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&23C8231&0&4030
Device ID PCI\VEN_1069&DEV_BA56&SUBSYS_00401069&REV_00\4&23C8231&0&4030
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 24
I/O Port 0x7000-0x7FFF
Driver c:\winnt\system32\drivers\dac2w2k.sys (185584, 9.00-04)

Name Adaptec AHA-2930CU PCI SCSI Controller
Caption Adaptec AHA-2930CU PCI SCSI Controller
Driver aic78xx
Status OK
PNP Device ID ROOT\LEGACY_AIC78XX\0000
Device ID ROOT\LEGACY_AIC78XX\0000
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
Driver c:\winnt\system32\drivers\aic78xx.sys (56848, v2.20b)

Name
Caption
Driver Not Available
Status OK
PNP Device ID ROOT\SCSIADAPTER\0000
Device ID ROOT\SCSIADAPTER\0000
Device Map Not Available
Index Not Available
Max Number Controlled Not Available

[Printing]

Name Port Name Server Name

No printing information

[Problem Devices]

Device PNP Device ID Error Code
PCI Memory Controller PCI\VEN_8086&DEV_1964&SUBSYS_10CE103C&REV_01\3&12F48E42&0&19 28

[USB]

Device PNP Device ID
No USB Devices

[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
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	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
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

APPENDIX C – TUNABLE PARAMETERS

atirage	atirage	c:\winnt\system32\drivers\atiragem.sys	Kernel Driver	True	Manual	Running	OK	
	Ignore	False	True					
atmarpc	ATM ARP Client Protocol	c:\winnt\system32\drivers\atmarpc.sys	Kernel Driver	False	Manual			
	Stopped	OK	Normal	False	False			
audstub	Audio Stub Driver	c:\winnt\system32\drivers\audstub.sys	Kernel Driver	True	Manual			
	Running	OK	Normal	False	True			
beep	Beep	c:\winnt\system32\drivers\beep.sys	Kernel Driver	True	System	Running	OK	
	Normal	False	True					
buslogic	BusLogic	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False							
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	False	Manual	Stopped	OK	
	Ignore	False	False					
cpqarray	Cpqarray	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False							
cpqarry2	cpqarry2	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False							
cpqfcalm	cpqfcalm	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False							
cpqfws2e	cpqfws2e	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False							
dac2w2k	dac2w2k	c:\winnt\system32\drivers\dac2w2k.sys	Kernel Driver	True	Boot	Running	OK	
	Normal	False	True					
dac960nt	dac960nt	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False							
deckzpsx	deckzpsx	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False							
dfsdriver	DfsDriver	c:\winnt\system32\drivers\dfs.sys	File System Driver	True	Boot			
	Running	OK	Normal	False	True			
disk	Disk Driver	c:\winnt\system32\drivers\disk.sys	Kernel Driver	True	Boot	Running	OK	
	Normal	False	True					
diskperf	Diskperf	c:\winnt\system32\drivers\diskperf.sys	Kernel Driver	False	Disabled	Stopped	OK	
	Normal	False	False					
dmboot	dmboot	c:\winnt\system32\drivers\dmboot.sys	Kernel Driver	False	Disabled	Stopped	OK	
	Normal	False	False					
dmio	Logical Disk Manager Driver	c:\winnt\system32\drivers\dmio.sys	Kernel Driver	True	Boot			
	Running	OK	Normal	False	True			
dmload	dmload	c:\winnt\system32\drivers\dmload.sys	Kernel Driver	True	Boot	Running	OK	
	Normal	False	True					
efs	EFS	c:\winnt\system32\drivers\efs.sys	File System Driver	True	Disabled			
	Running	OK	Normal	False	True			
fastfat	Fastfat	c:\winnt\system32\drivers\fastfat.sys	File System Driver	True	Disabled			
	Running	OK	Normal	False	True			
fd16_700	Fd16_700	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False							
fdc	Floppy Disk Controller Driver	c:\winnt\system32\drivers\fdc.sys	Kernel Driver	True				
	Manual	Running	OK	Normal	False	True		

fips	Fips	c:\winnt\system32\drivers\fips.sys	Kernel Driver	True	Auto	Running	OK	
	Normal	False	True					
fireport	fireport	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False							
flashpnt	flashpnt	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False							
flpydisk	Floppy Disk Driver	c:\winnt\system32\drivers\flpydisk.sys	Kernel Driver	True	Manual			
	Running	OK	Normal	False	True			
ftdisk	Volume Manager Driver	c:\winnt\system32\drivers\ftdisk.sys	Kernel Driver	True	Boot			
	Running	OK	Normal	False	True			
gamdrv	gamdrv	c:\winnt\system32\drivers\gamdrv.sys	Kernel Driver	True	Boot	Running	OK	
	Normal	False	True					
gnindis	cLAN NDIS Driver	c:\winnt\system32\drivers\gnindis.sys	Kernel Driver	True	Auto			
	Running	OK	Normal	False	True			
gnivia	cLAN VIA Driver	c:\winnt\system32\drivers\gnivia.sys	Kernel Driver	True	Auto			
	Running	OK	Normal	False	True			
gpc	Generic Packet Classifier	c:\winnt\system32\drivers\msgpc.sys	Kernel Driver	True	Manual			
	Running	OK	Normal	False	True			
hpn	Hewlett Packard NetRAID-4M RAID Driver				c:\winnt\system32\drivers\hpn.sys	Kernel Driver	False	
	Boot	Stopped	OK	Normal	False	False		
hptx	HP NetServer 10/100TX PCI LAN Adapter 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					
interruptaffinityfilter	Interrupt Affinity Filter		c:\winnt\system32\drivers\intfiltr.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			
kbdclass	Keyboard Class Driver	c:\winnt\system32\drivers\kbdclass.sys	Kernel Driver	True	System			
	Running	OK	Normal	False	True			
ksecdd	KSecDD	c:\winnt\system32\drivers\ksecdd.sys	Kernel Driver	True	Boot	Running	OK	
	Normal	False	True					
lbrtfdc	lbrtfdc	Not Available	Kernel Driver	False	System	Stopped	OK	Ignore	False
	False							
lp6nds35	lp6nds35	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False							
macdisk	macdisk	c:\winnt\system32\drivers\mac2w2k.sys	Kernel Driver	True	Boot	Running	OK	
	Normal	False	True					
mnmdd	mnmdd	c:\winnt\system32\drivers\mnmdd.sys	Kernel Driver	True	System	Running	OK	
	Ignore	False	True					
modem	Modem	c:\winnt\system32\drivers\modem.sys	Kernel Driver	False	Manual	Stopped	OK	
	Ignore	False	False					

APPENDIX C – TUNABLE PARAMETERS

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 class driver	c:\winnt\system32\drivers\parallel.sys	Kernel Driver	True	Manual				
	Running	OK	Normal	False	True				
parport	Parallel port driver	c:\winnt\system32\drivers\parport.sys	Kernel Driver	True	System				
	Running	OK	Ignore	False	True				
partmgr	PartMgr	c:\winnt\system32\drivers\partmgr.sys	Kernel Driver	True	Boot	Running	OK		
	Normal	False	True						
parvdm	ParVdm	c:\winnt\system32\drivers\parvdm.sys	Kernel Driver	True	Auto	Running	OK		
	Ignore	False	True						
pci	PCI Bus Driver	c:\winnt\system32\drivers\pci.sys	Kernel Driver	True	Boot				
	Running	OK	Critical	False	True				
pcidump	PCIDump	Not Available	Kernel Driver	False	System	Stopped	OK	Ignore	False
	False								
pciide	PCIIde	c:\winnt\system32\drivers\pciide.sys	Kernel Driver	True	Boot	Running	OK		
	Normal	False	True						
pcmcia	Pcmcia	c:\winnt\system32\drivers\pcmcia.sys	Kernel Driver	False	Disabled	Stopped	OK		
	Normal	False	False						
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				
sfloppy	Sfloppy	c:\winnt\system32\drivers\sfloppy.sys	Kernel Driver	False	System	Stopped	OK		
	Ignore	False	False						
sglfb	sglfb	c:\winnt\system32\drivers\sglfb.sys	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				

APPENDIX C – TUNABLE PARAMETERS

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	False
	Manual	Stopped	OK	Normal	False	False			
ultra66	ultra66	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
update	Microcode Update Driver		c:\winnt\system32\drivers\update.sys		Kernel Driver	True		Manual	
	Running	OK	Normal	False	True				
usbhub	Microsoft USB Standard Hub Driver				c:\winnt\system32\drivers\usbhub.sys			Kernel Driver	False
	Manual	Stopped	OK	Normal	False	False			
vgasave	VgaSave	c:\winnt\system32\drivers\vga.sys			Kernel Driver	True	System	Running	OK
	Ignore	False	True						
wanarp	Remote Access IP ARP Driver				c:\winnt\system32\drivers\wanarp.sys			Kernel Driver	True
	Manual	Running	OK	Normal	False	True			
wdica	WDICA	Not Available	Kernel Driver	False	Manual	Stopped	OK	Ignore	False
	False								

[Environment Variables]

Variable	Value	User Name
ComSpec	%SystemRoot%\system32\cmd.exe	<SYSTEM>
NUMBER_OF_PROCESSORS	6	<SYSTEM>
OS	Windows_NT	<SYSTEM>
Os2LibPath	%SystemRoot%\system32\os2\dll;	<SYSTEM>
Path	%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\system32\WBEM;C:\Program Files\Microsoft Visual Studio\VC98\Bin;C:\Program Files\Microsoft SQL Server\80\Tools\BINN	<SYSTEM>
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH	<SYSTEM>
PROCESSOR_ARCHITECTURE	x86	<SYSTEM>
PROCESSOR_IDENTIFIER	x86 Family 6 Model 10 Stepping 0, GenuineIntel	<SYSTEM>
PROCESSOR_LEVEL	6	<SYSTEM>
PROCESSOR_REVISION	0a00	<SYSTEM>
TEMP	%SystemRoot%\TEMP	<SYSTEM>
TMP	%SystemRoot%\TEMP	<SYSTEM>
windir	%SystemRoot%	<SYSTEM>
TEMP	%USERPROFILE%\Local Settings\Temp	PRF_SUT5\gamroot
TMP	%USERPROFILE%\Local Settings\Temp	PRF_SUT5\gamroot
TEMP	%USERPROFILE%\Local Settings\Temp	PRF_SUT5\Administrator
TMP	%USERPROFILE%\Local Settings\Temp	PRF_SUT5\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
No network connections information				

[Running Tasks]

Name	Path	Process ID	Priority	Min Working Set	Max Working Set	Start Time
	Version	Size	File Date			
system idle process		Not Available	0	0	Not Available	Not Available	Not Available	Unknown
	Unknown	Unknown						
system	Not Available	8	8	0	1413120	Not Available	Unknown	Unknown
	Unknown							
smss.exe	c:\winnt\system32\smss.exe		176	11	204800	1413120	8/8/2001 9:39:31
AM	5.00.2195.2901	44.27 KB (45,328 bytes)		6/12/2001 2:38:30 PM				
csrss.exe	Not Available	204	13	Not Available	Not Available	8/8/2001 9:39:36 AM		Unknown
	Unknown	Unknown						
winlogon.exe	c:\winnt\system32\winlogon.exe		224	13	204800	1413120	8/8/2001 9:39:38
AM	5.00.2195.2953	173.77 KB (177,936 bytes)		6/12/2001 2:41:06 PM				
services.exe	c:\winnt\system32\services.exe		252	9	204800	1413120	8/8/2001 9:39:39
AM	5.00.2195.2780	86.77 KB (88,848 bytes)		6/12/2001 2:38:27 PM				
lsass.exe	c:\winnt\system32\lsass.exe		264	9	204800	1413120	8/8/2001 9:39:39
AM	5.00.2195.2964	32.77 KB (33,552 bytes)		6/12/2001 2:37:40 PM				
gnconmgr.exe	c:\winnt\system32\gnconmgr.exe		408	8	204800	1413120		
	8/8/2001 9:39:42 AM	4.2.0.75	168.05 KB (172,079 bytes)		7/19/2001 11:12:58 AM			
svchost.exe	c:\winnt\system32\svchost.exe		468	8	204800	1413120	8/8/2001 9:39:44
AM	5.00.2134.1	7.77 KB (7,952 bytes)		12/7/1999 4:00:00 AM				
msdtc.exe	c:\winnt\system32\msdtc.exe		496	8	204800	1413120	8/8/2001 9:39:45
AM	1999.9.3421.3	6.77 KB (6,928 bytes)		3/6/2001 6:48:54 AM				
svchost.exe	c:\winnt\system32\svchost.exe		708	8	204800	1413120	8/8/2001 9:39:47
AM	5.00.2134.1	7.77 KB (7,952 bytes)		12/7/1999 4:00:00 AM				
snmp.exe	c:\winnt\system32\snmp.exe		740	8	204800	1413120	8/8/2001 9:39:47
AM	5.00.2173.1	29.77 KB (30,480 bytes)		3/6/2001 6:48:38 AM				

APPENDIX C – TUNABLE PARAMETERS

winmgmt.exe c:\winnt\system32\wbem\winmgmt.exe 788 8 204800 1413120 8/8/2001 9:39:48 AM 1.50.1085.0029 192.08 KB (196,685 bytes) 6/12/2001 2:41:08 PM
inetinfo.exe c:\winnt\system32\inetsrv\inetinfo.exe 844 8 204800 1413120 8/8/2001 9:39:52 AM 5.00.0984 14.27 KB (14,608 bytes) 6/12/2001 2:37:25 PM
svchost.exe c:\winnt\system32\svchost.exe 1132 8 204800 1413120 8/8/2001 9:40:14 AM 5.00.2134.1 7.77 KB (7,952 bytes) 12/7/1999 4:00:00 AM
explorer.exe c:\winnt\explorer.exe 752 8 204800 1413120 8/8/2001 9:41:00 AM 5.00.3315.2846 237.27 KB (242,960 bytes) 6/12/2001 2:37:15 PM
hpmon.exe c:\winnt\system32\hpmon.exe 248 8 204800 1413120 8/8/2001 9:41:01 AM 1.11 28.50 KB (29,184 bytes) 2/2/2001 9:16:12 AM
sqlmangr.exe c:\program files\microsoft sql server\80\tools\binn\sqlmangr.exe 1084 8 204800 1413120 8/8/2001 9:41:03 AM 2000.080.0382.00 72.57 KB (74,308 bytes) 6/19/2001 10:41:55 AM
mdm.exe c:\winnt\system32\mdm.exe 1052 8 204800 1413120 8/8/2001 9:41:04 AM 6.00.8424 121.29 KB (124,200 bytes) 3/6/2001 6:51:56 AM
mmc.exe c:\winnt\system32\mmc.exe 832 8 204800 1413120 8/8/2001 9:44:01 AM 5.00.2195.2301 589.27 KB (603,408 bytes) 6/12/2001 2:37:42 PM
rsvp.exe c:\winnt\system32\rsvp.exe 1388 8 204800 1413120 8/8/2001 9:45:25 AM 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 1.50.1085.0045 40.08 KB (41,040 bytes) 6/12/2001 2:39:33 PM 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 5.00.2195.2778 195.27 KB (199,952 bytes) 6/12/2001 2:36:54 PM Microsoft Corporation c:\winnt\system32\adsnt.dll
dbghelp.dll 5.00.2195.2104 159.27 KB (163,088 bytes) 6/12/2001 2:37:08 PM Microsoft Corporation c:\winnt\system32\dbghelp.dll
localsec.dll 5.00.2195.2130 230.27 KB (235,792 bytes) 6/12/2001 2:37:39 PM 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 5.00.2195.2165 287.27 KB (294,160 bytes) 6/12/2001 2:37:15 PM Microsoft Corporation c:\winnt\system32\filemgmt.dll
pdh.dll 5.00.2195.2739 147.77 KB (151,312 bytes) 6/12/2001 2:38:21 PM Microsoft Corporation c:\winnt\system32\pdh.dll
smlogcfg.dll 5.00.2195.2485 273.27 KB (279,824 bytes) 6/12/2001 2:38:30 PM 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) 5/4/2001 9:41:11 AM Microsoft Corporation c:\program files\common files\microsoft shared\msinfo\msinfo32.dll
riched20.dll 5.30.23.1205 421.27 KB (431,376 bytes) 6/12/2001 2:38:24 PM 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 5.00.2195.2104 41.77 KB (42,768 bytes) 6/12/2001 2:37:09 PM Executive Software International, Inc. c:\winnt\system32\dfrgsnap.dll
dmdskres.dll 2195.2104.297.3 119.50 KB (122,368 bytes) 6/12/2001 2:37:10 PM Microsoft Corp., VERITAS Software c:\winnt\system32\dmdskres.dll
dmutil.dll 2195.2104.297.3 42.27 KB (43,280 bytes) 6/12/2001 2:37:10 PM VERITAS Software Corp. c:\winnt\system32\dmutil.dll
ntmsapi.dll 5.00.1948.1 51.77 KB (53,008 bytes) 6/12/2001 2:38:17 PM Microsoft Corporation c:\winnt\system32\ntmsapi.dll
dmdskmgr.dll 2215.2215.297.3 160.27 KB (164,112 bytes) 6/12/2001 2:37:10 PM Microsoft Corp., VERITAS Software c:\winnt\system32\dmdskmgr.dll
mycomput.dll 5.00.2134.1 107.77 KB (110,352 bytes) 12/7/1999 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
mmc.exe 5.00.2195.2301 589.27 KB (603,408 bytes) 6/12/2001 2:37:42 PM Microsoft Corporation c:\winnt\system32\mmc.exe
msdbgen.dll 6.00.8146 9.83 KB (10,062 bytes) 5/25/1998 1:00:00 AM Microsoft Corporation c:\winnt\system32\msdbgen.dll
mdm.exe 6.00.8424 121.29 KB (124,200 bytes) 3/6/2001 6:51:56 AM Microsoft Corporation c:\winnt\system32\mdm.exe
sqlmangr.rll 2000.080.0194.00 96.00 KB (98,304 bytes) 6/19/2001 10:41:55 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) 6/19/2001 10:41:52 AM Microsoft Corporation c:\program files\microsoft sql server\80\tools\binn\resources\1033\sqlsvc.rll
sqlresld.dll 2000.080.0382.00 28.56 KB (29,248 bytes) 6/19/2001 10:41:52 AM Microsoft Corporation c:\program files\microsoft sql server\80\tools\binn\sqlresld.dll
odbcbcp.dll 2000.080.0380.00 28.57 KB (29,252 bytes) 6/19/2001 10:45:56 AM Microsoft Corporation c:\winnt\system32\odbcbcp.dll
sqlsvc.dll 2000.080.0382.00 92.56 KB (94,784 bytes) 6/19/2001 10:41:52 AM Microsoft Corporation c:\program files\microsoft sql server\80\tools\binn\sqlsvc.dll
w95scm.dll 2000.080.0194.00 48.56 KB (49,728 bytes) 6/19/2001 10:41:52 AM Microsoft Corporation c:\program files\microsoft sql server\80\tools\binn\w95scm.dll
sqlunirl.dll 2000.080.0380.00 176.56 KB (180,800 bytes) 4/9/2001 10:46:18 AM Microsoft Corporation c:\winnt\system32\sqlunirl.dll
sqlmangr.exe 2000.080.0382.00 72.57 KB (74,308 bytes) 6/19/2001 10:41:55 AM Microsoft Corporation c:\program files\microsoft sql server\80\tools\binn\sqlmangr.exe
hpmon.exe 1.11 28.50 KB (29,184 bytes) 2/2/2001 9:16:12 AM Hewlett-Packard Company c:\winnt\system32\hpmon.exe
thumbvw.dll 5.00.2920.0000 183.27 KB (187,664 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation c:\winnt\system32\thumbvw.dll
imm32.dll 5.00.2195.2821 94.27 KB (96,528 bytes) 6/12/2001 2:37:25 PM Microsoft Corporation c:\winnt\system32\imm32.dll
usp10.dll 1.0325.2195.2104 308.27 KB (315,664 bytes) 6/12/2001 2:39:13 PM Microsoft Corporation c:\winnt\system32\usp10.dll
msonsext.dll 9.0.3503 548.06 KB (561,210 bytes) 11/3/1999 4:38:34 PM Microsoft Corporation c:\progra~1\common~1\micros~1\webfol~1\msonsext.dll

APPENDIX C – TUNABLE PARAMETERS

wzshlstb.dll	3.0 (32-bit)	24.07 KB (24,644 bytes)	11/22/2000 8:00:00 AM	WinZip Computing, Inc.	c:\progra~1\winzip\wzshlstb.dll
mshtmled.dll	5.00.3315.2870	227.27 KB (232,720 bytes)	6/12/2001 2:37:56 PM	Microsoft Corporation	c:\winnt\system32\mshtmled.dll
imgutil.dll	5.00.3315.2870	30.77 KB (31,504 bytes)	6/12/2001 2:37:25 PM	Microsoft Corporation	c:\winnt\system32\imgutil.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
wininet.dll	5.00.3315.1000	456.77 KB (467,728 bytes)	6/12/2001 2:41:05 PM	Microsoft Corporation	c:\winnt\system32\wininet.dll
msdbg.dll	6.00.8424	67.50 KB (69,120 bytes)	3/6/2001 6:51:56 AM	Microsoft Corporation	c:\winnt\system32\msdbg.dll
shdoclc.dll	5.00.3315.2879	324.50 KB (332,288 bytes)	6/12/2001 2:38:28 PM	Microsoft Corporation	c:\winnt\system32\shdoclc.dll
pdm.dll	6.00.8424	179.27 KB (183,574 bytes)	3/6/2001 6:51:56 AM	Microsoft Corporation	c:\winnt\system32\pdm.dll
mshtml.dll	5.00.3315.2870	2.24 MB (2,345,232 bytes)	6/12/2001 2:37:55 PM	Microsoft Corporation	c:\winnt\system32\mshtml.dll
mlang.dll	5.00.3103.1000	510.77 KB (523,024 bytes)	6/12/2001 2:37:42 PM	Microsoft Corporation	c:\winnt\system32\mlang.dll
urlmon.dll	5.00.3315.1000	441.27 KB (451,856 bytes)	6/12/2001 2:39:12 PM	Microsoft Corporation	c:\winnt\system32\urlmon.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
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	5.00.3103.1000	13.27 KB (13,584 bytes)	6/12/2001 2:38:21 PM	Microsoft Corporation	c:\winnt\system32\powrprof.dll
batmeter.dll	5.00.3103.1000	20.27 KB (20,752 bytes)	6/12/2001 2:36:56 PM	Microsoft Corporation	c:\winnt\system32\batmeter.dll
stobject.dll	5.00.2195.2780	79.27 KB (81,168 bytes)	6/12/2001 2:38:37 PM	Microsoft Corporation	c:\winnt\system32\stobject.dll
webcheck.dll	5.00.3315.1000	251.77 KB (257,808 bytes)	6/12/2001 2:39:34 PM	Microsoft Corporation	c:\winnt\system32\webcheck.dll
msi.dll	1.11.2405.0	1.69 MB (1,767,184 bytes)	6/12/2001 2:37:56 PM	Microsoft Corporation	c:\winnt\system32\msi.dll
browselc.dll	5.00.3315.2846	34.50 KB (35,328 bytes)	6/12/2001 2:36:57 PM	Microsoft Corporation	c:\winnt\system32\browselc.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	5.00.3315.2846	788.77 KB (807,696 bytes)	6/12/2001 2:36:57 PM	Microsoft Corporation	c:\winnt\system32\browseui.dll
shdocvw.dll	5.00.3315.2879	1.05 MB (1,104,144 bytes)	6/12/2001 2:38:28 PM	Microsoft Corporation	c:\winnt\system32\shdocvw.dll
explorer.exe	5.00.3315.2846	237.27 KB (242,960 bytes)	6/12/2001 2:37:15 PM	Microsoft Corporation	c:\winnt\explorer.exe
tapisrv.dll	5.00.2195.2955	169.27 KB (173,328 bytes)	6/12/2001 2:38:39 PM	Microsoft Corporation	c:\winnt\system32\tapisrv.dll
aqueue.dll	5.00.0984	312.77 KB (320,272 bytes)	6/12/2001 2:36:55 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\aqueue.dll
seo.dll	5.00.0984	229.27 KB (234,768 bytes)	6/12/2001 2:38:27 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\seo.dll
iscomlog.dll	5.00.0984	24.77 KB (25,360 bytes)	6/12/2001 2:37:34 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\iscomlog.dll
lonsint.dll	5.00.0984	11.77 KB (12,048 bytes)	6/12/2001 2:37:40 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\lonsint.dll
inetsloc.dll	5.00.0984	20.27 KB (20,752 bytes)	6/12/2001 2:37:25 PM	Microsoft Corporation	c:\winnt\system32\inetsloc.dll
staxmem.dll	5.00.0984	8.27 KB (8,464 bytes)	5/4/2001 9:37:00 AM	Microsoft Corporation	c:\winnt\system32\staxmem.dll
exstrace.dll	5.00.0984	13.77 KB (14,096 bytes)	5/4/2001 9:37:04 AM	Microsoft Corporation	c:\winnt\system32\exstrace.dll
rwnh.dll	5.00.0984	10.77 KB (11,024 bytes)	3/6/2001 6:51:42 AM	Microsoft Corporation	c:\winnt\system32\rwnh.dll
fcachdll.dll	5.00.0984	43.77 KB (44,816 bytes)	3/6/2001 6:51:41 AM	Microsoft Corporation	c:\winnt\system32\fcachdll.dll
iisfecnv.dll	5.00.0984	7.27 KB (7,440 bytes)	5/4/2001 9:37:02 AM	Microsoft Corporation	c:\winnt\system32\inetsrv\iisfecnv.dll
isatq.dll	5.00.0984	60.27 KB (61,712 bytes)	6/12/2001 2:37:34 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\isatq.dll
infocomm.dll	5.00.0984	238.27 KB (243,984 bytes)	6/12/2001 2:37:25 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\infocomm.dll
smtpsvc.dll	5.00.0984	420.27 KB (430,352 bytes)	6/12/2001 2:38:32 PM	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)	6/12/2001 2:38:37 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\svcext.dll
admexs.dll	5.00.0984	27.77 KB (28,432 bytes)	6/12/2001 2:35:36 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\admexs.dll
wamreg.dll	5.00.0984	45.77 KB (46,864 bytes)	6/12/2001 2:39:24 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\wamreg.dll
metadata.dll	5.00.0984	68.77 KB (70,416 bytes)	6/12/2001 2:37:42 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\metadata.dll
iismap.dll	5.00.0984	55.77 KB (57,104 bytes)	6/12/2001 2:37:25 PM	Microsoft Corporation	c:\winnt\system32\iismap.dll
nsepm.dll	5.00.0984	43.27 KB (44,304 bytes)	6/12/2001 2:38:09 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\nsepm.dll
admwprox.dll	5.00.0984	31.77 KB (32,528 bytes)	5/4/2001 9:37:04 AM	Microsoft Corporation	c:\winnt\system32\admwprox.dll
coadmin.dll	5.00.0984	39.27 KB (40,208 bytes)	6/12/2001 2:37:07 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\coadmin.dll
iisadmin.dll	5.00.0984	15.27 KB (15,632 bytes)	6/12/2001 2:37:25 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\iisadmin.dll
rpcref.dll	5.00.0984	4.27 KB (4,368 bytes)	6/12/2001 2:38:24 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\rpcref.dll
inetinfo.exe	5.00.0984	14.27 KB (14,608 bytes)	6/12/2001 2:37:25 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\inetinfo.exe
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
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
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
ntmarta.dll 5.00.2195.2862 98.77 KB (101,136 bytes) 6/12/2001 2:38:17 PM Microsoft Corporation
c:\winnt\system32\ntmarta.dll
provthrd.dll 1.50.1085.0000 68.07 KB (69,708 bytes) 5/4/2001 9:40:56 AM 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
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 1.50.1085.0038 1.02 MB (1,073,232 bytes) 6/12/2001 2:36:59 PM Microsoft Corporation
c:\winnt\system32\wbem\cimwin32.dll
wbemsvc.dll 1.50.1085.0007 40.07 KB (41,036 bytes) 6/12/2001 2:39:33 PM Microsoft Corporation
c:\winnt\system32\wbem\wbemsvc.dll
wbemess.dll 1.50.1085.0039 364.07 KB (372,804 bytes) 6/12/2001 2:39:33 PM Microsoft Corporation
c:\winnt\system32\wbem\wbemess.dll
fastprox.dll 1.50.1085.0037 144.08 KB (147,536 bytes) 6/12/2001 2:37:15 PM Microsoft Corporation
c:\winnt\system32\wbem\fastprox.dll
wbemcore.dll 1.50.1085.0036 628.07 KB (643,140 bytes) 6/12/2001 2:39:28 PM Microsoft Corporation
c:\winnt\system32\wbem\wbemcore.dll
wbemcomn.dll 1.50.1085.0021 692.07 KB (708,675 bytes) 6/12/2001 2:39:25 PM Microsoft Corporation c:\winnt\system32\wbem\wbemcomn.dll
winmgmt.exe 1.50.1085.0029 192.08 KB (196,685 bytes) 6/12/2001 2:41:08 PM Microsoft Corporation
c:\winnt\system32\wbem\winmgmt.exe
gamagent.dll 4.00-00 240.62 KB (246,392 bytes) 7/13/2001 10:18:56 AM Mylex Corporation
c:\winnt\system32\gamserv\gamagent.dll
gnisnmp.dll Not Available 48.00 KB (49,152 bytes) 7/9/2001 4:59:03 PM Not Available c:\winnt\system32\gnisnmp.dll
odbcint.dll 3.520.7326.0 88.00 KB (90,112 bytes) 6/19/2001 10:45:51 AM Microsoft Corporation
c:\winnt\system32\odbcint.dll
comdlg32.dll 5.00.3103.1000 236.77 KB (242,448 bytes) 6/12/2001 2:37:07 PM Microsoft Corporation
c:\winnt\system32\comdlg32.dll
odbc32.dll 3.520.7326.0 216.27 KB (221,456 bytes) 6/19/2001 10:45:49 AM Microsoft Corporation
c:\winnt\system32\odbc32.dll
sqlsnmp.dll 2000.080.0194.00 60.00 KB (61,440 bytes) 6/19/2001 10:41:08 AM Microsoft Corporation
c:\program files\microsoft sql server\mssql\binn\sqlsnmp.dll
ftpmib.dll 5.00.0984 6.27 KB (6,416 bytes) 6/12/2001 2:37:20 PM Microsoft Corporation
c:\winnt\system32\inetsrv\ftpmib.dll
iisrtl.dll 5.00.0984 119.77 KB (122,640 bytes) 6/12/2001 2:37:25 PM Microsoft Corporation
c:\winnt\system32\iisrtl.dll
infoadmn.dll 5.00.0984 13.27 KB (13,584 bytes) 6/12/2001 2:37:25 PM Microsoft Corporation
c:\winnt\system32\infoadmn.dll
httpmib.dll 5.00.0984 9.27 KB (9,488 bytes) 6/12/2001 2:37:22 PM Microsoft Corporation
c:\winnt\system32\inetsrv\httpmib.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
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
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
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
evntagnt.dll 5.00.2167.1 94.77 KB (97,040 bytes) 3/6/2001 6:48:38 AM Microsoft Corporation
c:\winnt\system32\evntagnt.dll
snmpmib.dll 5.00.2134.1 5.77 KB (5,904 bytes) 3/6/2001 6:48:38 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) 3/6/2001 6:48:38 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) 3/6/2001 6:48:38 AM Microsoft Corporation
c:\winnt\system32\snmp.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 5.00.2195.2779 457.27 KB (468,240 bytes) 6/12/2001 2:38:04 PM Microsoft Corporation
c:\winnt\system32\netshell.dll
netman.dll 5.00.2195.2779 89.27 KB (91,408 bytes) 6/12/2001 2:38:04 PM Microsoft Corporation
c:\winnt\system32\netman.dll
ntmsdba.dll 5.00.2195.2779 167.27 KB (171,280 bytes) 6/12/2001 2:38:17 PM Microsoft Corporation
c:\winnt\system32\ntmsdba.dll
rasdlg.dll 5.00.2195.2671 514.27 KB (526,608 bytes) 6/12/2001 2:38:23 PM Microsoft Corporation
c:\winnt\system32\rasdlg.dll
netcfgx.dll 5.00.2195.2228 534.77 KB (547,600 bytes) 6/12/2001 2:38:04 PM Microsoft Corporation
c:\winnt\system32\netcfgx.dll
rasmans.dll 5.00.2195.2728 147.27 KB (150,800 bytes) 6/12/2001 2:38:23 PM Microsoft Corporation
c:\winnt\system32\rasmans.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
ntmssvc.dll 5.00.2195.2779 391.27 KB (400,656 bytes) 6/12/2001 2:38:17 PM Microsoft Corporation
c:\winnt\system32\ntmssvc.dll
es.dll 2000.2.3471.1 222.27 KB (227,600 bytes) 6/12/2001 2:37:14 PM Microsoft Corporation
c:\winnt\system32\es.dll
mtxoci.dll 2000.2.3471.1 101.77 KB (104,208 bytes) 6/12/2001 2:38:03 PM Microsoft Corporation
c:\winnt\system32\mtxoci.dll
resutils.dll 5.00.2195.2787 39.77 KB (40,720 bytes) 6/12/2001 2:38:24 PM Microsoft Corporation
c:\winnt\system32\resutils.dll
clusapi.dll 5.00.2195.2104 54.27 KB (55,568 bytes) 6/12/2001 2:37:06 PM 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) 3/6/2001 6:48:55 AM Microsoft Corporation
c:\winnt\system32\xolehlp.dll
msdtclog.dll 1999.9.3421.3 89.77 KB (91,920 bytes) 3/6/2001 6:48:54 AM Microsoft Corporation
c:\winnt\system32\msdtclog.dll
mtxclu.dll 2000.2.3471.1 51.27 KB (52,496 bytes) 6/12/2001 2:38:03 PM Microsoft Corporation
c:\winnt\system32\mtxclu.dll

APPENDIX C – TUNABLE PARAMETERS

msdtcprx.dll 2000.2.3471.1 665.77 KB (681,744 bytes) 6/12/2001 2:37:52 PM Microsoft Corporation
c:\winnt\system32\msdtcprx.dll
txfaux.dll 2000.2.3471.1 374.27 KB (383,248 bytes) 6/12/2001 2:39:08 PM Microsoft Corporation
c:\winnt\system32\txfaux.dll
msdtctm.dll 2000.2.3471.1 1.07 MB (1,120,528 bytes) 6/12/2001 2:37:52 PM Microsoft Corporation
c:\winnt\system32\msdtctm.dll
msdtc.exe 1999.9.3421.3 6.77 KB (6,928 bytes) 3/6/2001 6:48:54 AM Microsoft Corporation
c:\winnt\system32\msdtc.exe
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
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
rnr20.dll 5.00.2195.2871 35.77 KB (36,624 bytes) 6/12/2001 2:38:24 PM Microsoft Corporation
c:\winnt\system32\rnr20.dll
wshtcpip.dll 5.00.2195.2104 17.27 KB (17,680 bytes) 6/12/2001 2:41:12 PM Microsoft Corporation
c:\winnt\system32\wshtcpip.dll
msafd.dll 5.00.2195.2779 106.77 KB (109,328 bytes) 6/12/2001 2:37:48 PM Microsoft Corporation
c:\winnt\system32\msafd.dll
rpcss.dll 5.00.2195.2815 231.27 KB (236,816 bytes) 6/12/2001 2:38:24 PM 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
vipl.dll 4.2.0.75 80.00 KB (81,920 bytes) 7/19/2001 11:12:58 AM Giganet Incorporated
c:\winnt\system32\vipl.dll
gnconmgr.exe 4.2.0.75 168.05 KB (172,079 bytes) 7/19/2001 11:12:58 AM Giganet Incorporated
c:\winnt\system32\gnconmgr.exe
dssenh.dll 5.00.2195.2228 142.77 KB (146,192 bytes) 6/12/2001 2:38:05 PM Microsoft Corporation
c:\winnt\system32\dssenh.dll
scecli.dll 5.00.2195.2780 105.27 KB (107,792 bytes) 6/12/2001 2:38:27 PM 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 5.00.2195.2778 130.77 KB (133,904 bytes) 6/12/2001 2:36:58 PM Microsoft Corporation
c:\winnt\system32\certcli.dll
mswsock.dll 5.00.2195.2871 62.77 KB (64,272 bytes) 6/12/2001 2:38:02 PM Microsoft Corporation
c:\winnt\system32\mswsock.dll
ntdsatq.dll 5.00.2195.2878 31.27 KB (32,016 bytes) 6/12/2001 2:38:14 PM Microsoft Corporation
c:\winnt\system32\ntdsatq.dll
ntdsa.dll 5.00.2195.2899 990.77 KB (1,014,544 bytes) 6/12/2001 2:38:13 PM Microsoft
Corporation c:\winnt\system32\ntdsa.dll
kdcsvc.dll 5.00.2195.2878 137.77 KB (141,072 bytes) 6/12/2001 2:37:35 PM 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 5.00.2195.2671 21.27 KB (21,776 bytes) 6/12/2001 2:38:24 PM Microsoft Corporation
c:\winnt\system32\rassfm.dll
mpr.dll 5.00.2195.2779 53.27 KB (54,544 bytes) 6/12/2001 2:37:43 PM Microsoft Corporation
c:\winnt\system32\mpr.dll
rsabase.dll 5.00.2195.2228 128.27 KB (131,344 bytes) 6/12/2001 2:38:24 PM Microsoft Corporation
c:\winnt\system32\rsabase.dll
schannel.dll 5.00.2195.2922 138.27 KB (141,584 bytes) 6/12/2001 2:38:27 PM Microsoft Corporation
c:\winnt\system32\schannel.dll
netlogon.dll 5.00.2195.2865 357.77 KB (366,352 bytes) 6/12/2001 2:38:04 PM Microsoft Corporation
c:\winnt\system32\netlogon.dll
kerberos.dll 5.00.2195.2913 198.77 KB (203,536 bytes) 6/12/2001 2:37:35 PM 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 5.00.2195.2918 369.77 KB (378,640 bytes) 6/12/2001 2:38:27 PM Microsoft Corporation
c:\winnt\system32\samsrv.dll
lsasrv.dll 5.00.2195.2964 492.77 KB (504,592 bytes) 6/12/2001 2:37:40 PM Microsoft Corporation
c:\winnt\system32\lsasrv.dll
lsass.exe 5.00.2195.2964 32.77 KB (33,552 bytes) 6/12/2001 2:37:40 PM Microsoft Corporation
c:\winnt\system32\lsass.exe
ntlsapi.dll 5.00.2134.1 6.77 KB (6,928 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\ntlsapi.dll
wmicore.dll 5.00.2195.2842 72.27 KB (74,000 bytes) 6/12/2001 2:41:11 PM Microsoft Corporation
c:\winnt\system32\wmicore.dll
browser.dll 5.00.2195.2778 48.27 KB (49,424 bytes) 6/12/2001 2:36:57 PM Microsoft Corporation
c:\winnt\system32\browser.dll
esent.dll 6.0.3940.13 1.08 MB (1,135,376 bytes) 6/12/2001 2:37:14 PM Microsoft Corporation
c:\winnt\system32\esent.dll
psbase.dll 5.00.2195.2779 111.77 KB (114,448 bytes) 6/12/2001 2:38:22 PM 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 5.00.2195.2780 95.27 KB (97,552 bytes) 6/12/2001 2:41:10 PM Microsoft Corporation
c:\winnt\system32\wkssvc.dll
srvsvc.dll 5.00.2195.2904 79.27 KB (81,168 bytes) 6/12/2001 2:38:36 PM 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 2195.2778.297.3 11.77 KB (12,048 bytes) 6/12/2001 2:37:10 PM VERITAS Software Corp.
c:\winnt\system32\dmserver.dll
winsta.dll 5.00.2195.2386 36.77 KB (37,648 bytes) 6/12/2001 2:41:10 PM Microsoft Corporation
c:\winnt\system32\winsta.dll
lmhsvc.dll 5.00.2195.2778 9.77 KB (10,000 bytes) 6/12/2001 2:37:39 PM Microsoft Corporation
c:\winnt\system32\lmhsvc.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 5.00.2195.2780 54.77 KB (56,080 bytes) 6/12/2001 2:38:23 PM Microsoft Corporation
c:\winnt\system32\rasman.dll
rasapi32.dll 5.00.2195.2671 189.77 KB (194,320 bytes) 6/12/2001 2:38:23 PM 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 5.00.2195.2842 127.27 KB (130,320 bytes) 6/12/2001 2:36:54 PM Microsoft Corporation
c:\winnt\system32\adsldpc.dll
activeds.dll 5.00.2195.2778 174.77 KB (178,960 bytes) 6/12/2001 2:35:36 PM Microsoft Corporation
c:\winnt\system32\activeds.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
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
dhcpcsvc.dll 5.00.2195.2778 88.77 KB (90,896 bytes) 6/12/2001 2:37:09 PM Microsoft Corporation
c:\winnt\system32\dhcpcsvc.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

APPENDIX C – TUNABLE PARAMETERS

ntdsapi.dll 5.00.2195.2661 55.77 KB (57,104 bytes) 6/12/2001 2:38:14 PM Microsoft Corporation
c:\winnt\system32\ntdsapi.dll
scesrv.dll 5.00.2195.2780 226.27 KB (231,696 bytes) 6/12/2001 2:38:27 PM 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 5.00.2195.2780 86.77 KB (88,848 bytes) 6/12/2001 2:38:27 PM Microsoft Corporation
c:\winnt\system32\services.exe
msv1_0.dll 5.00.2195.2900 111.77 KB (114,448 bytes) 6/12/2001 2:38:02 PM Microsoft Corporation
c:\winnt\system32\msv1_0.dll
clbcatq.dll 2000.2.3471.1 496.77 KB (508,688 bytes) 6/12/2001 2:37:06 PM Microsoft Corporation
c:\winnt\system32\clbcatq.dll
oleaut32.dll 2.40.4517 612.27 KB (626,960 bytes) 6/12/2001 2:38:20 PM Microsoft Corporation
c:\winnt\system32\oleaut32.dll
cscui.dll 5.00.2195.2959 228.27 KB (233,744 bytes) 6/12/2001 2:37:08 PM Microsoft Corporation
c:\winnt\system32\cscui.dll
winspool.drv 5.00.2195.2780 109.77 KB (112,400 bytes) 6/12/2001 2:41:08 PM 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 5.00.2195.2780 53.77 KB (55,056 bytes) 6/12/2001 2:41:11 PM Microsoft Corporation
c:\winnt\system32\wlnotify.dll
cscdll.dll 5.00.2195.2401 98.27 KB (100,624 bytes) 6/12/2001 2:37:08 PM 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
rsaenh.dll 5.00.2195.2228 130.77 KB (133,904 bytes) 6/12/2001 2:38:06 PM Microsoft Corporation
c:\winnt\system32\rsaenh.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 5.00.2195.2887 969.77 KB (993,040 bytes) 6/12/2001 2:38:19 PM Microsoft Corporation
c:\winnt\system32\ole32.dll
imagehlp.dll 5.00.2195.2778 125.77 KB (128,784 bytes) 6/12/2001 2:37:25 PM 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 5.131.2195.2833 451.27 KB (462,096 bytes) 6/12/2001 2:37:08 PM Microsoft Corporation
c:\winnt\system32\crypt32.dll
wintrust.dll 5.131.2195.2779 162.27 KB (166,160 bytes) 6/12/2001 2:41:10 PM Microsoft Corporation
c:\winnt\system32\wintrust.dll
setupapi.dll 5.00.2195.2663 555.77 KB (569,104 bytes) 6/12/2001 2:38:28 PM 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 5.81 537.77 KB (550,672 bytes) 6/12/2001 2:37:07 PM Microsoft Corporation
c:\winnt\system32\comctl32.dll
shlwapi.dll 5.00.3315.1000 282.77 KB (289,552 bytes) 6/12/2001 2:38:30 PM Microsoft Corporation
c:\winnt\system32\shlwapi.dll
shell32.dll 5.00.3315.2902 2.25 MB (2,359,056 bytes) 6/12/2001 2:38:29 PM Microsoft Corporation
c:\winnt\system32\shell32.dll
msgina.dll 5.00.2195.2779 324.27 KB (332,048 bytes) 6/12/2001 2:37:53 PM Microsoft Corporation
c:\winnt\system32\msgina.dll
wsock32.dll 5.00.2195.2871 21.27 KB (21,776 bytes) 6/12/2001 2:41:12 PM Microsoft Corporation
c:\winnt\system32\wsock32.dll

dnsapi.dll 5.00.2195.2785 130.77 KB (133,904 bytes) 6/12/2001 2:37:10 PM Microsoft Corporation
c:\winnt\system32\dnsapi.dll
wldap32.dll 5.00.2195.2797 125.27 KB (128,272 bytes) 6/12/2001 2:41:11 PM 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 5.00.2195.2780 67.77 KB (69,392 bytes) 6/12/2001 2:41:12 PM Microsoft Corporation
c:\winnt\system32\ws2_32.dll
samlib.dll 5.00.2195.2780 49.77 KB (50,960 bytes) 6/12/2001 2:38:25 PM 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 5.00.2195.2808 303.77 KB (311,056 bytes) 6/12/2001 2:38:03 PM 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 5.00.2195.2862 46.77 KB (47,888 bytes) 6/12/2001 2:38:27 PM Microsoft Corporation
c:\winnt\system32\secur32.dll
sfc.dll 5.00.2195.2896 92.11 KB (94,320 bytes) 6/12/2001 2:38:28 PM 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 5.00.2195.2780 361.77 KB (370,448 bytes) 6/12/2001 2:39:13 PM Microsoft Corporation
c:\winnt\system32\userenv.dll
user32.dll 5.00.2195.2821 392.77 KB (402,192 bytes) 6/12/2001 2:39:13 PM Microsoft Corporation
c:\winnt\system32\user32.dll
gdi32.dll 5.00.2195.2778 228.77 KB (234,256 bytes) 6/12/2001 2:37:20 PM Microsoft Corporation
c:\winnt\system32\gdi32.dll
rpcrt4.dll 5.00.2195.2832 437.27 KB (447,760 bytes) 6/12/2001 2:38:24 PM Microsoft Corporation
c:\winnt\system32\rpcrt4.dll
advapi32.dll 5.00.2195.2867 351.77 KB (360,208 bytes) 6/12/2001 2:36:54 PM Microsoft Corporation
c:\winnt\system32\advapi32.dll
kernel32.dll 5.00.2195.2778 714.77 KB (731,920 bytes) 6/12/2001 2:37:35 PM Microsoft Corporation
c:\winnt\system32\kernel32.dll
msvcrt.dll 6.10.8924.0 284.05 KB (290,869 bytes) 6/12/2001 2:38:02 PM Microsoft Corporation
c:\winnt\system32\msvcrt.dll
winlogon.exe 5.00.2195.2953 173.77 KB (177,936 bytes) 6/12/2001 2:41:06 PM Microsoft Corporation
c:\winnt\system32\winlogon.exe
sfcfiles.dll 5.00.2195.2967 948.27 KB (971,024 bytes) 6/12/2001 2:38:28 PM Microsoft Corporation
c:\winnt\system32\sfcfiles.dll
ntdll.dll 5.00.2195.2779 478.77 KB (490,256 bytes) 6/12/2001 2:38:13 PM Microsoft Corporation
c:\winnt\system32\ntdll.dll
smss.exe 5.00.2195.2901 44.27 KB (45,328 bytes) 6/12/2001 2:38:30 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	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	Running	Auto	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					

APPENDIX C – TUNABLE PARAMETERS

Indexing Service cisvc Stopped Manual Share Process c:\winnt\system32\cisvc.exe
Normal LocalSystem 0
ClipBook ClipSrv Stopped Manual Own Process c:\winnt\system32\clipsrv.exe Normal
LocalSystem 0
Distributed File System Dfs Stopped Manual Own Process c:\winnt\system32\dfssvc.exe
Normal LocalSystem 0
DHCP Client Dhcp Running Auto Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0
Logical Disk Manager Administrative Service dmadmin Stopped Manual Share Process
c:\winnt\system32\dmadmin.exe /com Normal LocalSystem 0
Logical Disk Manager dmserver Running Auto Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0
DNS Client Dnscache Stopped Manual Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0
Event Log Eventlog Running Auto Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0
COM+ Event System EventSystem Running Manual Share Process c:\winnt\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Fax Service Fax Stopped Manual Own Process c:\winnt\system32\faxsvc.exe Normal
LocalSystem 0
Mylex Global Array Manager Server gamscm Stopped Manual Own Process system32\gamserv\gamscm.exe
Normal LocalSystem 0
cLAN Connection Manager GniConMgr Running Auto Own Process c:\winnt\system32\gnconmgr.exe
Normal LocalSystem 0
IIS Admin Service IISADMIN Running Auto Share Process
c:\winnt\system32\inetsrv\inetinfo.exe Normal LocalSystem 0
Intersite Messaging IsmServ Stopped Disabled Own Process c:\winnt\system32\ismserv.exe
Normal LocalSystem 0
Kerberos Key Distribution Center kdc Stopped Disabled Share Process
c:\winnt\system32\lsass.exe Normal LocalSystem 0
Server lanmanserver Running Auto Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0
Workstation lanmanworkstation Running Auto Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0
License Logging Service LicenseService Stopped Manual 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 Manual 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 1
FTP Publishing Service MSFTPSVC Stopped Manual 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
MSSQLSERVER MSSQLSERVER Stopped Manual Own Process
c:\progra~1\micros~4\mssql\binn\sqlservr.exe Normal LocalSystem 0
MSSQLServerADHelper MSSQLServerADHelper Stopped Manual Own Process c:\program files\microsoft sql
server\80\tools\binn\sqladhlp.exe Normal LocalSystem 0
Network DDE NetDDE Stopped Manual Share Process c:\winnt\system32\netdde.exe
Normal LocalSystem 0
Network DDE DSDM NetDDEdsdm Stopped Manual Share Process
c:\winnt\system32\netdde.exe Normal LocalSystem 0
Net Logon Netlogon Stopped Manual Share Process c:\winnt\system32\lsass.exe
Normal LocalSystem 0
Network Connections Netman Running Manual Share Process c:\winnt\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
File Replication NtFrs Stopped Manual Own Process c:\winnt\system32\ntfrs.exe Ignore
LocalSystem 0
NT LM Security Support Provider NtLmSsp Stopped Manual Share Process
c:\winnt\system32\lsass.exe Normal LocalSystem 0
Removable Storage NtmsSvc Running Auto Share Process c:\winnt\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Plug and Play PlugPlay Running Auto Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0
IPSEC Policy Agent PolicyAgent Stopped Manual Share Process c:\winnt\system32\lsass.exe
Normal LocalSystem 0
Protected Storage ProtectedStorage Running Auto Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Remote Access Auto Connection Manager RasAuto Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Remote Access Connection Manager RasMan Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Routing and Remote Access RemoteAccess Stopped Disabled Share Process
c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Remote Registry Service RemoteRegistry Stopped Manual Own Process c:\winnt\system32\regsvc.exe
Normal LocalSystem 0
Remote Procedure Call (RPC) Locator RpcLocator Stopped Manual Own Process c:\winnt\system32\locator.exe
Normal LocalSystem 0
Remote Procedure Call (RPC) RpcSs Running Auto Share Process
c:\winnt\system32\svchost -k rpcss Normal LocalSystem 0
QoS RSVP RSVP Running Manual Own Process c:\winnt\system32\rsvp.exe -s Normal
LocalSystem 0
Security Accounts Manager SamSs Running Auto Share Process
c:\winnt\system32\lsass.exe Normal LocalSystem 0
Smart Card Helper SCardDrv Stopped Manual Share Process c:\winnt\system32\scardsvr.exe
Ignore LocalSystem 0
Smart Card SCardSvr Stopped Manual Share Process c:\winnt\system32\scardsvr.exe Ignore
LocalSystem 0
Task Scheduler Schedule 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 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 Manual Own Process c:\winnt\system32\spoolsv.exe Normal
LocalSystem 0
SQLSERVERAGENT SQLSERVERAGENT Stopped Manual Own Process
c:\progra~1\micros~4\mssql\binn\sqlagent.exe Normal .\Administrator 0
Performance Logs and Alerts SysmonLog Stopped Auto 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 Manual Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Uninterruptible Power Supply UPS Stopped Manual Own Process c:\winnt\system32\ups.exe
Normal LocalSystem 0
Utility Manager UtilMan Stopped Manual Own Process c:\winnt\system32\utilman.exe
Normal LocalSystem 0
Visual Studio Analyzer RPC bridge Visual Studio Analyzer RPC bridge Stopped Manual Own Process
c:\program files\microsoft visual studio\common\tools\vs-ent98\vanalyzr\varpc.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 Management Instrumentation Driver Extensions Wmi Running Manual Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0

[Program Groups]

Group Name Name User Name
Accessories Default User:Accessories Default User
Accessories\Accessibility Default User:Accessories\Accessibility Default User
Accessories\Entertainment Default User:Accessories\Entertainment Default User
Accessories\System Tools Default User:Accessories\System Tools Default User
Startup Default User:Startup Default User
Accessories All Users:Accessories All Users
Accessories\Accessibility All Users:Accessories\Accessibility All Users
Accessories\Communications All Users:Accessories\Communications All Users
Accessories\Entertainment All Users:Accessories\Entertainment All Users
Accessories\Games All Users:Accessories\Games All Users
Accessories\Microsoft Script Debugger All Users:Accessories\Microsoft Script Debugger All Users
Accessories\System Tools All Users:Accessories\System Tools All Users
Administrative Tools All Users:Administrative Tools All Users
GigaNet All Users:GigaNet All Users
Microsoft SQL Server All Users:Microsoft SQL Server All Users
Microsoft Visual Studio 6.0 All Users:Microsoft Visual Studio 6.0 All Users
Microsoft Visual Studio 6.0\Microsoft Visual SourceSafe All Users:Microsoft Visual Studio 6.0\Microsoft Visual
SourceSafe All Users
Microsoft Visual Studio 6.0\Microsoft Visual Studio 6.0 Enterprise Tools All Users:Microsoft Visual Studio 6.0\Microsoft
Visual Studio 6.0 Enterprise Tools 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
Startup All Users:Startup All Users
VNC All Users:VNC All Users
VNC\Administrative Tools All Users:VNC\Administrative Tools All Users
WinZip All Users:WinZip All Users
Accessories PRF_SUT5\gamroot:Accessories PRF_SUT5\gamroot
Accessories\Accessibility PRF_SUT5\gamroot:Accessories\Accessibility PRF_SUT5\gamroot
Accessories\Entertainment PRF_SUT5\gamroot:Accessories\Entertainment PRF_SUT5\gamroot
Accessories\System Tools PRF_SUT5\gamroot:Accessories\System Tools PRF_SUT5\gamroot
Startup PRF_SUT5\gamroot:Startup PRF_SUT5\gamroot
Accessories PRF_SUT5\Administrator:Accessories PRF_SUT5\Administrator
Accessories\Accessibility PRF_SUT5\Administrator:Accessories\Accessibility PRF_SUT5\Administrator
Accessories\Entertainment PRF_SUT5\Administrator:Accessories\Entertainment PRF_SUT5\Administrator
Accessories\System Tools PRF_SUT5\Administrator:Accessories\System Tools PRF_SUT5\Administrator
Administrative Tools PRF_SUT5\Administrator:Administrative Tools PRF_SUT5\Administrator
Microsoft Web Publishing PRF_SUT5\Administrator:Microsoft Web Publishing PRF_SUT5\Administrator
Startup PRF_SUT5\Administrator:Startup PRF_SUT5\Administrator

[Startup Programs]

Program Command User Name Location
cacheMAC cachemac.cmd All Users Common Startup
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 mspaint.exe

[Internet Explorer 5]

[Following are sub-categories of this main category]

[Summary]

Item Value
Version 5.00.3315.1000
Build 53315.1000
Product ID 51879-OEM-0000941-93697
Application Path C:\Program Files\Internet Explorer
Language English (United States)
Active Printer Not Available

Cipher Strength 168-bit
Content Advisor Disabled
IEAK Install No

[File Versions]

File Version Size Date Path Company
advapi32.dll 5.0.2195.2867 352 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft
Corporation

APPENDIX C – TUNABLE PARAMETERS

advpack.dll	5.0.3103.1000	87 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
browselc.dll	5.0.3315.2846	35 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
browseui.dll	5.0.3315.2846	789 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
ckcnv.exe	5.0.2189.1	9 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation	
comctl32.dll	5.81.3103.1000	538 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
crypt32.dll	5.131.2195.2833	451 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
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	5.0.3103.1000	57 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
iexplore.exe	5.0.2920.0	59 KB	12/7/1999 5:00:00 AM	C:\Program Files\Internet Explorer	Microsoft Corporation	
imagehlp.dll	5.0.2195.2778	126 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
imghelp.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available	
inseng.dll	5.0.3103.1000	72 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
jobexec.dll	5.0.0.1	47 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation	
jscript.dll	5.1.0.5907	476 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
jsproxy.dll	5.0.2920.0	13 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation	
msaahtml.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available	
mshtml.dll	5.0.3315.2870	2290 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
msjava.dll	5.0.3802.0	923 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
msoss.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available	
msxml.dll	8.0.5718.1	493 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
occache.dll	5.0.3103.1000	86 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
ole32.dll	5.0.2195.2887	970 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
oleaut32.dll	2.40.4517.0	612 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
olepro32.dll	5.0.4517.0	160 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
rsabase.dll	5.0.2195.2228	128 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
rsaenh.dll	5.0.2195.2228	131 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
rsapi32.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available	
rsasig.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available	
schannel.dll	5.1.2195.0	138 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
shdoc401.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available	
shdocvw.dll	5.0.3315.2879	1078 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
shell32.dll	5.0.3315.2902	2304 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
shlwapi.dll	5.0.3315.1000	283 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
url.dll	5.0.2920.0	82 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation	
urlmon.dll	5.0.3315.1000	441 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
vbscript.dll	5.1.0.5907	428 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
webcheck.dll	5.0.3315.1000	252 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
win.com	5.0.2134.1	24 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation	
wininet.dll	5.0.3315.1000	457 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
winsock.dll	3.10.0.103	3 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation	
wintrust.dll	5.131.2195.2779	162 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
wsock.vxd	<File Missing>	Not Available	Not Available	Not Available	Not Available	
wsock32.dll	5.0.2195.2871	21 KB	5/4/2001 12:05:02 PM	C:\WINNT\system32	Microsoft Corporation	
wsock32n.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available	

[Connectivity]

Item	Value
Connection Preference	Never dial
EnableHttp1.1	1
ProxyHttp1.1	0

LAN Settings

AutoConfigProxy	wininet.dll
AutoProxyDetectMode	Enabled
AutoConfigURL	
Proxy	Enabled
ProxyServer	web-proxy.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	4590 MB
Maximum Cache Size	270 MB
Available Cache Size	258 MB

[List of Objects]

Program File Status	CodeBase
CV3 Class Installed	http://windowsupdate.microsoft.com/R731/V31Controls/x86/nt5/en/actsetup.cab

[Content]

[Following are sub-categories of this main category]

[Summary]

Item	Value
Content Advisor	Disabled

APPENDIX C – TUNABLE PARAMETERS

[Personal Certificates]

Issued To	Issued By	Validity	Signature Algorithm
Administrator	Administrator	3/6/2001 to 2/10/2101	sha1RSA

[Other People Certificates]

Issued To	Issued By	Validity	Signature Algorithm
No other people certificate information available			

[Publishers]

Name
No publisher information available

[Security]

Zone	Security Level
Local intranet	Medium-low
Trusted sites	Low
Internet	Medium
Restricted sites	High

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	-2147483648	2147483647	63	63
allow updates	0	1	0	0
awe enabled	0	1	1	1
c2 audit mode	0	1	0	0
cost threshold for parallelism	0	32767	5	5
cursor threshold	-1	2147483647	-1	-1
default full-text language	0	2147483647	1033	1033
default language	0	9999	0	0
fill factor (%)	0	100	0	0
index create memory (KB)	704	2147483647	0	0
lightweight pooling	0	1	1	1
locks	5000	2147483647	0	0
max degree of parallelism	0	32	1	1
max server memory (MB)	4	2147483647	2147483647	2147483647
max text repl size (B)	0	2147483647	65536	65536
max worker threads	32	32767	238	238
media retention	0	365	0	0
min memory per query (KB)	512	2147483647	1024	1024
min server memory (MB)	0	2147483647	0	0
nested triggers	0	1	1	1
network packet size (B)	512	65536	4096	4096
open objects	0	2147483647	0	0
priority boost	0	1	1	1
query governor cost limit	0	2147483647	0	0
query wait (s)	-1	2147483647	-1	-1
recovery interval (min)	0	32767	56	56
remote access	0	1	1	1
remote login timeout (s)	0	2147483647	20	20
remote proc trans	0	1	0	0
remote query timeout (s)	0	2147483647	20	20
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 Mylex Configuration Parameters

Mylex Device Drivers and Firmware

The following device drivers were added:
Mylex BIOS: 6.01-01
Mylex Frimware: 6.00-02
Miniport Driver: 9.00-04 (dac2w2k.sys)
Filtered Driver: 9.00-04 (mac2w2k.sys)

Mylex Registry Key

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dac2w2k\Parameters\Device]
"DriverParameter"="ConfigureSIR=19"

C.9 Client System Configuration Parameters

COM+ Settings

TPCC.AllTxns:
Activation:
Enable Object Pooling selected
Minimum Pool Size: 70
Maximum Pool Size: 70
Creation Timeout: 60000
Enable Object Construction
Enable Just In Time Activation
Concurrency:
Concurrency Required

Microsoft IIS Registry Parameters

Key Name: SYSTEM\CurrentControlSet\Services\InetInfo
Class Name: <NO CLASS>
Last Write Time: 3/28/2001 - 4:53 PM

Key Name: SYSTEM\CurrentControlSet\Services\InetInfo\Parameters
Class Name: <NO CLASS>
Last Write Time: 6/15/2001 - 11:32 AM
Value 0
Name: DispatchEntries
Type: REG_MULTI_SZ
Data: LDAPSVC
SMTPSVC

Value 1
Name: ListenBackLog
Type: REG_DWORD
Data: 0x19

Value 2
Name: PoolThreadLimit
Type: REG_DWORD
Data: 0x3fe

Value 3
Name: ThreadTimeout
Type: REG_DWORD
Data: 0x15180

Key Name: SYSTEM\CurrentControlSet\Services\InetInfo\Performance
Class Name: <NO CLASS>
Last Write Time: 8/16/2001 - 3:33 PM
Value 0
Name: Close
Type: REG_SZ
Data: CloseINFOPerformanceData

Value 1
Name: Collect
Type: REG_SZ
Data: CollectINFOPerformanceData

Value 2
Name: First Counter
Type: REG_DWORD
Data: 0x802

Value 3
Name: First Help
Type: REG_DWORD
Data: 0x803

Value 4
Name: Last Counter
Type: REG_DWORD
Data: 0x842

Value 5
Name: Last Help
Type: REG_DWORD
Data: 0x843

Value 6
Name: Library
Type: REG_SZ
Data: infoctrs.dll

Value 7
Name: Library Validation Code
Type: REG_BINARY
Data:
00000000 94 cb 3f de 9e b7 c0 01 - 10 25 00 00 00 00 00 00 .Ë?Þ.·À..%.....

Value 8
Name: Open
Type: REG_SZ
Data: OpenINFOPerformanceData

Value 9
Name: WbemAdapFileSize
Type: REG_DWORD
Data: 0x2510

Value 10
Name: WbemAdapFileTime
Type: REG_BINARY
Data:
00000000 00 43 02 1f cd d4 c0 01 - .C..ÍÔÀ.

Value 11
Name: WbemAdapStatus
Type: REG_DWORD
Data: 0

World Wide Web Service Registry Parameters

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC
Class Name: <NO CLASS>
Last Write Time: 5/18/2001 - 12:07 PM
Value 0
Name: DependOnGroup
Type: REG_MULTI_SZ
Data:

Value 1
Name: DependOnService
Type: REG_MULTI_SZ
Data: IISADMIN

Value 2
Name: Description
Type: REG_SZ
Data: Provides Web connectivity and administration through the Internet Information Services snap-in.

Value 3
Name: DisplayName
Type: REG_SZ
Data: World Wide Web Publishing Service

Value 4
Name: ErrorControl
Type: REG_DWORD
Data: 0x1

Value 5
Name: FailureActions
Type: REG_BINARY
Data:
00000000 ff ff ff ff 00 00 00 00 - 00 00 00 00 03 00 00 00 ÿÿÿÿ.....
00000010 70 57 0c 00 00 00 00 00 - 00 00 00 00 00 00 00 00 pW.....
00000020 00 00 00 00 00 00 00 00 - 00 00 00 00 .....

Value 6
Name: ImagePath
Type: REG_EXPAND_SZ
Data: C:\WINNT\System32\inetsrv\inetinfo.exe

Value 7
Name: ObjectName
Type: REG_SZ
Data: LocalSystem

Value 8
Name: Start
Type: REG_DWORD
Data: 0x2

Value 9
Name: Type
Type: REG_DWORD
Data: 0x20

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\ASP
Class Name: <NO CLASS>
Last Write Time: 3/28/2001 - 4:54 PM
Value 0
Name: NOTE
Type: REG_SZ
Data: This is for backward compatibility only.

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\ASP\Parameters
Class Name: <NO CLASS>
Last Write Time: 3/28/2001 - 4:54 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Enum
Class Name: <NO CLASS>
Last Write Time: 8/16/2001 - 3:31 PM
Value 0
Name: 0

APPENDIX C – TUNABLE PARAMETERS

Type: REG_SZ
Data: Root\LEGACY_W3SVC\0000

Value 1
Name: Count
Type: REG_DWORD
Data: 0x1

Value 2
Name: NextInstance
Type: REG_DWORD
Data: 0x1

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters
Class Name: <NO CLASS>
Last Write Time: 3/28/2001 - 5:36 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: 3/28/2001 - 4:55 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory
Class Name: <NO CLASS>
Last Write Time: 3/28/2001 - 4:55 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory
Class Name: <NO CLASS>
Last Write Time: 3/28/2001 - 4:55 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map
Class Name: <NO CLASS>
Last Write Time: 3/28/2001 - 5:08 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots
Class Name: <NO CLASS>
Last Write Time: 8/16/2001 - 12:47 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: /D
Type: REG_SZ
Data: d:\,,217

Value 3
Name: /IISAdmin
Type: REG_SZ
Data: C:\WINNT\System32\inetsrv\iisadmin,,201

Value 4
Name: /IISHelp
Type: REG_SZ
Data: c:\winnt\help\iishelp,,201

Value 5
Name: /IISSamples
Type: REG_SZ
Data: c:\inetpub\iissamples,,201

Value 6
Name: /MSADC
Type: REG_SZ
Data: c:\program files\common files\system\msadc,,205

Value 7
Name: /PBSData
Type: REG_SZ
Data: C:\Program Files\Phone Book Service\Data,,1

APPENDIX C – TUNABLE PARAMETERS

```
Value 8
  Name:         /PBServer
  Type:         REG_SZ
  Data:         C:\Program Files\Phone Book Service\Bin,,5

Value 9
  Name:         /Printers
  Type:         REG_SZ
  Data:         C:\WINNT\web\printers,,201

Value 10
  Name:         /Scripts
  Type:         REG_SZ
  Data:         c:\inetpub\scripts,,204


Key Name:       SYSTEM\CurrentControlSet\Services\W3SVC\Performance
Class Name:     <NO CLASS>
Last Write Time:  8/16/2001 - 3:33 PM
Value 0
  Name:         Close
  Type:         REG_SZ
  Data:         CloseW3PerformanceData

Value 1
  Name:         Collect
  Type:         REG_SZ
  Data:         CollectW3PerformanceData

Value 2
  Name:         First Counter
  Type:         REG_DWORD
  Data:         0x844

Value 3
  Name:         First Help
  Type:         REG_DWORD
  Data:         0x845

Value 4
  Name:         Last Counter
  Type:         REG_DWORD
  Data:         0x8e6

Value 5
  Name:         Last Help
  Type:         REG_DWORD
  Data:         0x8e7

Value 6
  Name:         Library
  Type:         REG_SZ
  Data:         w3ctrs.dll

Value 7
  Name:         Library Validation Code
  Type:         REG_BINARY
  Data:
00000000  c0 d1 0f e1 9e b7 c0 01 - 10 3d 00 00 00 00 00 00  ÀÑ.á.·À..=......

Value 8
  Name:         Open
  Type:         REG_SZ
  Data:         OpenW3PerformanceData

Value 9
  Name:         WbemAdapFileSize
  Type:         REG_DWORD
  Data:         0x1d10

Value 10
  Name:         WbemAdapFileTime
  Type:         REG_BINARY
  Data:
00000000  00 43 02 1f cd d4 c0 01 -                    .C..ÍÔÀ.

Value 11
  Name:         WbemAdapStatus
  Type:         REG_DWORD
  Data:         0


Key Name:       SYSTEM\CurrentControlSet\Services\W3SVC\Security
Class Name:     <NO CLASS>
Last Write Time:  3/28/2001 - 4:54 PM
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 74 00 6f 00  ............t.o.
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 - 72 00 73 00 00 00 18 00   ... ...r.s.....
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 72 00 73 00  .... ...#...r.s.
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 -                        ........
```

TPCC Application Registry Parameters

```
Key Name:       SOFTWARE\Microsoft\TPCC
Class Name:     <NO CLASS>
Last Write Time:  7/31/2001 - 12:32 PM
Value 0
  Name:         COM_SinglePool
  Type:         REG_SZ
```

APPENDIX C – TUNABLE PARAMETERS

Data: YES

Value 1
Name: DB_Protocol
Type: REG_SZ
Data: ODBC

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_sut5

Value 5
Name: DbUser
Type: REG_SZ
Data: sa

Value 6
Name: MaxConnections
Type: REG_DWORD
Data: 0x4e20

Value 7
Name: MaxPendingDeliveries
Type: REG_DWORD
Data: 0x3e8

Value 8
Name: NumberOfDeliveryThreads
Type: REG_DWORD
Data: 0x7

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: 08/08/2001 09:28:29 AM
[System Information]

[Following are sub-categories of this main category]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Server
Version	5.0.2195 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	N2_CLIENT1
System Manufacturer	Hewlett-Packard
System Model	HP NetServer
System Type	X86-based PC
Processor	x86 Family 6 Model 8 Stepping 6 GenuineIntel ~867 Mhz
BIOS Version	01/23/01
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition1
Locale	United States
User Name	N2_CLIENT1\Administrator
Time Zone	Pacific Daylight Time
Total Physical Memory	523,760 KB
Available Physical Memory	416,700 KB
Total Virtual Memory	2,039,704 KB
Available Virtual Memory	1,850,064 KB
Page File Space	1,515,944 KB
Page File	C:\pagefile.sys

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

Resource	Device
No conflicted/shared resources	

[DMA]

Channel	Device	Status
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
0x0D00-0x1BFF	PCI bus	OK

APPENDIX C – TUNABLE PARAMETERS

0x2100-0xFFFF	PCI bus	OK		
0x1800-0x183F	HP NetServer 10/100TX PCI LAN Adapter			OK
0x1400-0x14FF	ATI Technologies Inc. RAGE XL PCI	OK		
0x03B0-0x03BB	ATI Technologies Inc. RAGE XL PCI	OK		
0x03C0-0x03DF	ATI Technologies Inc. RAGE XL PCI	OK		
0x0A79-0x0A79	ISAPNP Read Data Port	OK		
0x0279-0x0279	ISAPNP Read Data Port	OK		
0x02F4-0x02F7	ISAPNP Read Data Port	OK		
0x0081-0x008F	Direct memory access controller	OK		
0x00C0-0x00DF	Direct memory access controller	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		
0x0096-0x0097	Motherboard resources	OK		
0x0F50-0x0F58	Motherboard resources	OK		
0x1100-0x111F	Motherboard resources	OK		
0x1200-0x121F	Motherboard resources	OK		
0x0060-0x0060	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard			OK
0x0064-0x0064	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard			OK
0x0070-0x0071	System CMOS/real time clock	OK		
0x03F8-0x03FF	Communications Port (COM1)	OK		
0x02F8-0x02FF	Communications Port (COM2)	OK		
0x0378-0x037F	Printer Port (LPT1)	OK		
0x03F0-0x03F5	Standard floppy disk controller	OK		
0x03F7-0x03F7	Standard floppy disk controller	OK		
0x1840-0x184F	Standard Dual Channel PCI IDE Controller			OK
0x01F0-0x01F7	Primary IDE Channel	OK		
0x03F6-0x03F6	Primary IDE Channel	OK		
0x0170-0x0177	Secondary IDE Channel	OK		
0x0376-0x0376	Secondary IDE Channel	OK		
0x1C00-0x20FF	PCI bus	OK		
0x1C00-0x20FF	Symbios Logic 896, 22910 PCI SCSI Adapter			OK
0x2000-0x20FF	Symbios Logic 896, 22910 PCI SCSI Adapter			OK

[IRQs]

IRQ Number	Device
9	Microsoft ACPI-Compliant System
20	HP NetServer 10/100TX PCI LAN Adapter
13	Numeric data processor
1	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
12	PS/2 Compatible Mouse
8	System CMOS/real time clock
4	Communications Port (COM1)
3	Communications Port (COM2)
6	Standard floppy disk controller
14	Primary IDE Channel
5	Standard OpenHCD USB Host Controller
21	cLAN Host Adapter
24	Symbios Logic 896, 22910 PCI SCSI Adapter
25	Symbios Logic 896, 22910 PCI SCSI Adapter

[Memory]

Range	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	ATI Technologies Inc. RAGE XL PCI	OK
0xC0000-0xC3FFF	PCI bus	OK
0xC4000-0xC7FFF	PCI bus	OK
0xC8000-0xCBFFF	PCI bus	OK
0xCC000-0xCFFFF	PCI bus	OK
0xD0000-0xD3FFF	PCI bus	OK
0xD4000-0xD7FFF	PCI bus	OK
0xD8000-0xDBFFF	PCI bus	OK
0xDC000-0xDFFFF	PCI bus	OK
0xFA000000-0xFB10FFFF	PCI bus	OK
0xFA000000-0xFB10FFFF	ATI Technologies Inc. RAGE XL PCI	OK
0xFB100000-0xFB100FFF	HP NetServer 10/100TX PCI LAN Adapter	OK
0xFB000000-0xFB0FFFFF	HP NetServer 10/100TX PCI LAN Adapter	OK
0xFB101000-0xFB101FFF	ATI Technologies Inc. RAGE XL PCI	OK
0xFFFD0000-0xFFFEFFFF	System board	OK
0xFB102000-0xFB102FFF	Standard OpenHCD USB Host Controller	OK
0xFB110000-0xFCFFFFFF	PCI bus	OK
0xFB110000-0xFCFFFFFF	cLAN Host Adapter	OK
0xFB120000-0xFB13FFFF	cLAN Host Adapter	OK
0xFB200000-0xFB3FFFFF	cLAN Host Adapter	OK
0xFC000000-0xFCFFFFFF	cLAN Host Adapter	OK
0xFB142000-0xFB1423FF	Symbios Logic 896, 22910 PCI SCSI Adapter	OK
0xFB140000-0xFB141FFF	Symbios Logic 896, 22910 PCI SCSI Adapter	OK
0xFB142400-0xFB1427FF	Symbios Logic 896, 22910 PCI SCSI Adapter	OK
0xFB144000-0xFB145FFF	Symbios Logic 896, 22910 PCI SCSI 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)		12/7/1999 4:00:00 AM
c:\winnt\system32\msg723.acm	Microsoft Corporation		OK				
		C:\WINNT\System32\MSG723.ACM		4.4.3385	106.77 KB (109,328 bytes)		2/15/2001 3:11:20 PM
c:\winnt\system32\lhacm.acm	Microsoft Corporation		OK				
		C:\WINNT\System32\LHACM.ACM		4.4.3385	33.27 KB (34,064 bytes)		2/15/2001 3:11:20 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\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\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
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
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

[Video Codecs]

Codec	Manufacturer	Description	Status	File	Version	Size	Creation Date

APPENDIX C – TUNABLE PARAMETERS

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) 2/15/2001 3:11:20 PM
c:\winnt\system32\msh263.drv Microsoft Corporation OK
C:\WINNT\System32\MSH263.DRV 4.4.3385 252.27 KB (258,320 bytes) 2/15/2001 3:10:50 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	TEAC CD-540E
Manufacturer	(Standard CD-ROM drives)
Status	OK
Transfer Rate	Not Available
SCSI Target ID	0
PNP Device ID	IDE\CDROMTEAC_CD-540E_____1.0A___\5&F3420B7&0&0.0.0

[Sound Device]

Item	Value
No sound devices	

[Display]

Item	Value
Name	ATI Technologies Inc. RAGE XL PCI
PNP Device ID	PCI\VEN_1002&DEV_4752&SUBSYS_10E1103C&REV_65\3&13C0B0C5&0&38
Adapter Type	ATI RAGE XL PCI (B22), ATI Technologies Inc. compatible
Adapter Description	ATI Technologies Inc. RAGE XL PCI
Adapter RAM	4.00 MB (4,194,304 bytes)
Installed Drivers	atidrab.dll
Driver Version	5.00.2179.1
INF File	display.inf (atirage3 section)
Color Planes	1
Color Table Entries	4294967296
Resolution	1024 x 768 x 75 hertz
Bits/Pixel	32

[Infrared]

Item	Value
No infrared devices	

[Input]

[Following are sub-categories of this main category]

[Keyboard]

Item	Value
Description	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
Name	Enhanced (101- or 102-key)
Layout	00000409
PNP Device ID	ACPI\PNP0303\5&2C631CF0&0
NumberOfFunctionKeys	12

[Pointing Device]

Item	Value
Hardware Type	PS/2 Compatible Mouse
Number of Buttons	3
Status	OK
PNP Device ID	ACPI\PNP0F13\5&2C631CF0&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 categ ory]

[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_10CA103C&REV_08\3&13C0B0C5&0&10
Last Reset	8/8/2001 2:25:10 AM
Index	0
Service Name	HPTX
IP Address	200.20.1.200
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:E0:18:C4:83:15
Service Name	HPTX
IRQ Number	20
I/O Port	0x1800-0x183F
Driver	c:\winnt\system32\drivers\hptxnt5.sys (85264, 4.02.27.0000)

APPENDIX C – TUNABLE PARAMETERS

Name [00000001] RAS Async Adapter
Adapter Type Not Available
Product Name RAS Async Adapter
Installed True
PNP Device ID Not Available
Last Reset 8/8/2001 2:25:10 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 8/8/2001 2:25:10 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 8/8/2001 2:25:10 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 8/8/2001 2:25:10 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 8/8/2001 2:25:10 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 8/8/2001 2:25:10 AM
Index 6
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name NdisWan
Driver c:\winnt\system32\drivers\ndiswan.sys (90768, 5.00.2184.1)

Name [00000007] HP NetServer 10/100TX PCI LAN Adapter
Adapter Type Not Available

APPENDIX C – TUNABLE PARAMETERS

Product Name HP NetServer 10/100TX PCI LAN Adapter
Installed True
PNP Device ID Not Available
Last Reset 8/8/2001 2:25:10 AM
Index 7
Service Name HPTX
IP Address 200.20.1.200
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled True
DHCP Server 255.255.255.255
DHCP Lease Expires 7/17/2001 7:02:07 PM
DHCP Lease Obtained 7/17/2001 6:02:07 PM
MAC Address 00:E0:18:C4:83:15
Service Name Not Available

Name [00000008] cLAN Host Adapter
Adapter Type Ethernet 802.3
Product Name cLAN Host Adapter
Installed True
PNP Device ID PCI\VEN_135B&DEV_0001&SUBSYS_00000000&REV_00\3&1070020&0&10
Last Reset 8/8/2001 2:25:10 AM
Index 8
Service Name GNINDIS
IP Address 16.75.206.191
IP Subnet 255.255.248.0
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:FA:00:0F:13
Service Name GNINDIS
IRQ Number 21
Driver c:\winnt\system32\drivers\gnindis.sys (22726, 4.2.0.75)

[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_{920DF390-1CCA-46B2-9BDB-EC12EE3AD9C0}] SEQPACKET 4
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True

APPENDIX C – TUNABLE PARAMETERS

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_{920DF390-1CCA-46B2-9BDB-EC12EE3AD9C0}] DATAGRAM 4
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{FAA18BC5-2C0A-4FDF-86BE-ACFE53199EDF}] SEQPACKET 3
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{FAA18BC5-2C0A-4FDF-86BE-ACFE53199EDF}] DATAGRAM 3
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{ABA15699-958A-49C1-8DF6-6359AE77F507}] 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_{ABA15699-958A-49C1-8DF6-6359AE77F507}] 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_{45C78637-2678-430C-B4C7-54172510BEEA}] 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

APPENDIX C – TUNABLE PARAMETERS

SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{45C78637-2678-430C-B4C7-54172510BEEA}] 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_{BA162956-BF1F-4E4C-AFD1-B8D8E8E54D31}] 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_{BA162956-BF1F-4E4C-AFD1-B8D8E8E54D31}] 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	Not Available
Maximum Output Buffer Size	Not Available
Settable Baud Rate	Not Available
Settable Data Bits	Not Available
Settable Flow Control	Not Available
Settable Parity	Not Available
Settable Parity Check	Not Available
Settable Stop Bits	Not Available
Settable RLSD	Not Available
Supports RLSD	Not Available
Supports 16 Bit Mode	Not Available
Supports Special Characters	Not Available
Baud Rate	9600
Bits/Byte	8
Stop Bits	1
Parity	None
Busy	-1
Abort Read/Write on Error	Not Available
Binary Mode Enabled	Not Available
Continue XMit on XOff	Not Available
CTS Outflow Control	Not Available
Discard NULL Bytes	Not Available
DSR Outflow Control	Not Available
DSR Sensitivity	Not Available
DTR Flow Control Type	Not Available
EOF Character	Not Available
Error Replace Character	Not Available
Error Replacement Enabled	Not Available
Event Character	Not Available
Parity Check Enabled	-1
RTS Flow Control Type	Not Available
XOff Character	19
XOffXMit Threshold	512

APPENDIX C – TUNABLE PARAMETERS

XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control Not Available
XOnXOff OutFlow Control Not Available
IRQ Number 4
I/O Port 0x03F8-0x03FF
Driver c:\winnt\system32\drivers\serial.sys (62448, 5.00.2134.1)

Name COM2
Status OK
PNP Device ID ACPI\PNP0501\2
Maximum Input Buffer Size 0
Maximum Output Buffer Size False
Settable Baud Rate True
Settable Data Bits True
Settable Flow Control True
Settable Parity True
Settable Parity Check True
Settable Stop Bits True
Settable RLSD True
Supports RLSD True
Supports 16 Bit Mode False
Supports Special Characters False
Baud Rate 9600
Bits/Byte 8
Stop Bits 1
Parity None
Busy 0
Abort Read/Write on Error 0
Binary Mode Enabled -1
Continue XMit on XOff 0
CTS Outflow Control 0
Discard NULL Bytes 0
DSR Outflow Control 0
DSR Sensitivity 0
DTR Flow Control Type Enable
EOF Character 0
Error Replace Character 0
Error Replacement Enabled 0
Event Character 0
Parity Check Enabled 0
RTS Flow Control Type Enable
XOff Character 19
XOffXMit Threshold 512
XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control 0
IRQ Number 3
I/O Port 0x02F8-0x02FF
Driver c:\winnt\system32\drivers\serial.sys (62448, 5.00.2134.1)

[Parallel]

Item Value
Name LPT1
PNP Device ID ACPI\PNP0400\1

[Storage]

[Following are sub-categories of this main category]

[Drives]

Item Value
Drive A:
Description 3 1/2 Inch Floppy Drive

Drive C:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 8.46 GB (9,088,901,120 bytes)
Free Space 4.42 GB (4,746,199,040 bytes)
Volume Name
Volume Serial Number 9495570C
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 C 68-F511 SCSI Disk Device
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 1
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 2
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 9097159680 bytes
Drive TotalCylinders 1106
Drive TotalSectors 17767890
Drive TotalTracks 282030
Drive TracksPerCylinder 255

[SCSI]

Item Value
Name 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_60E0103C&REV_07\3&1070020&0&28
Device ID PCI\VEN_1000&DEV_000B&SUBSYS_60E0103C&REV_07\3&1070020&0&28
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 24
I/O Port 0x1C00-0x20FF
Driver c:\winnt\system32\drivers\sym_hi.sys (21136, 5.00.2134.1)

APPENDIX C – TUNABLE PARAMETERS

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_60E0103C&REV_07\3&1070020&0&29
Device ID PCI\VEN_1000&DEV_000B&SUBSYS_60E0103C&REV_07\3&1070020&0&29
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 25
I/O Port 0x2000-0x20FF
Driver c:\winnt\system32\drivers\sym_hi.sys (21136, 5.00.2134.1)

[Printing]

Name	Port Name	Server Name		
Acrobat PDFWriter	LPT1:	Not Available		
Acrobat Distiller	C:\Program Files\Adobe\Acrobat 4.0\PDF Output*.pdf	Not Available		
\\TPC_E60\HP LaserJet 6P/6MP - Enhanced	LPT1:	\\TPC_E60		

[Problem Devices]

Device	PNP Device ID	Error Code
No Problem Devices		

[USB]

Device	PNP Device ID
Standard OpenHCD USB Host Controller	PCI\VEN_1166&DEV_0220&SUBSYS_02201166&REV_04\3&13C0B0C5&0&7A
USB Root Hub	USB\ROOT_HUB\4&6EF0984&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								
aic116x	aic116x	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
aic78u2	aic78u2	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
aic78xx	aic78xx	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
ami0nt	ami0nt	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
amsint	amsint	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
asc	asc	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
asc3350p	asc3350p	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
asc3550	asc3550	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
asyncmac	RAS Asynchronous Media Driver		c:\winnt\system32\drivers\asyncmac.sys		Kernel Driver	False			
	Manual	Stopped	OK	Normal	False	False			
atapi	Standard IDE/ESDI Hard Disk Controller		c:\winnt\system32\drivers\atapi.sys		Kernel Driver	True	Boot		
	Running	OK	Normal	False	True				
atdisk	Atdisk	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Ignore	False
	False								
atirage3	atirage3	c:\winnt\system32\drivers\atimpab.sys		Kernel Driver	True	Manual	Running	OK	
	Ignore	False	True						
atmarpc	ATM ARP Client Protocol	c:\winnt\system32\drivers\atmarpc.sys		Kernel Driver	False	Manual			
	Stopped	OK	Normal	False	False				
audstub	Audio Stub Driver		c:\winnt\system32\drivers\audstub.sys		Kernel Driver	True	Manual		
	Running	OK	Normal	False	True				
beep	Beep	c:\winnt\system32\drivers\beep.sys		Kernel Driver	True	System	Running	OK	
	Normal	False	True						
buslogic	BusLogic	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
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								
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						

APPENDIX C – TUNABLE PARAMETERS

dmboot	dmboot	c:\winnt\system32\drivers\dmboot.sys	Kernel Driver	False	Disabled	Stopped	OK		
	Normal	False	False						
dmio	Logical Disk Manager Driver		c:\winnt\system32\drivers\dmio.sys	Kernel Driver	True	Boot			
	Running	OK	Normal	False	True				
dmload	dmload	c:\winnt\system32\drivers\dmload.sys	Kernel Driver	True	Boot	Running	OK		
	Normal	False	True						
efs	EFS	c:\winnt\system32\drivers\efs.sys	File System Driver	True	Disabled				
	Running	OK	Normal	False	True				
fastfat	Fastfat	c:\winnt\system32\drivers\fastfat.sys	File System Driver	True	Disabled				
	Running	OK	Normal	False	True				
fd16_700	Fd16_700	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
fdc	Floppy Disk Controller Driver		c:\winnt\system32\drivers\fdc.sys	Kernel Driver	True				
	Manual	Running	OK	Normal	False	True			
fireport	fireport	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
flashpnt	flashpnt	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
flpydisk	Floppy Disk Driver		c:\winnt\system32\drivers\flpydisk.sys	Kernel Driver	True	Manual			
	Running	OK	Normal	False	True				
ftdisk	Volume Manager Driver		c:\winnt\system32\drivers\ftdisk.sys	Kernel Driver	True	Boot			
	Running	OK	Normal	False	True				
gnindis	cLAN NDIS Driver		c:\winnt\system32\drivers\gnindis.sys	Kernel Driver	True	Auto			
	Running	OK	Normal	False	True				
gnivia	cLAN VIA Driver		c:\winnt\system32\drivers\gnivia.sys	Kernel Driver	True	Auto			
	Running	OK	Normal	False	True				
gpc	Generic Packet Classifier		c:\winnt\system32\drivers\msgpc.sys	Kernel Driver	True	Manual			
	Running	OK	Normal	False	True				
hptx	HP NetServer 10/100TX PCI LAN Adapter 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	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
ipfilterdriver	IP Traffic Filter Driver		c:\winnt\system32\drivers\ipfltdrv.sys	Kernel Driver	False	Manual			
	Stopped	OK	Normal	False	False				
ipinip	IP in IP Tunnel Driver		c:\winnt\system32\drivers\ipinip.sys	Kernel Driver	False	Manual			
	Stopped	OK	Normal	False	False				
ipnat	IP Network Address Translator		c:\winnt\system32\drivers\ipnat.sys	Kernel Driver	False				
	Manual	Stopped	OK	Normal	False	False			
ipsec	IPSEC driver	c:\winnt\system32\drivers\ipsec.sys	Kernel Driver	True	Manual	Running	OK		
	Normal	False	True						
ipsraidn	ipsraidn	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
isapnp	PnP ISA/EISA Bus Driver		c:\winnt\system32\drivers\isapnp.sys	Kernel Driver	True	Boot			
	Running	OK	Critical	False	True				
kbdclass	Keyboard Class Driver		c:\winnt\system32\drivers\kbdclass.sys	Kernel Driver	True	System			
	Running	OK	Normal	False	True				
ksecdd	KSecDD	c:\winnt\system32\drivers\ksecdd.sys	Kernel Driver	True	Boot	Running	OK		
	Normal	False	True						
lbrtfdc	lbrtfdc	Not Available	Kernel Driver	False	System	Stopped	OK	Ignore	False
	False								
lp6nds35	lp6nds35	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
mnmdd	mnmdd	c:\winnt\system32\drivers\mnmdd.sys	Kernel Driver	True	System	Running	OK		
	Ignore	False	True						
modem	Modem	c:\winnt\system32\drivers\modem.sys	Kernel Driver	False	Manual	Stopped	OK		
	Ignore	False	False						
mouclass	Mouse Class Driver		c:\winnt\system32\drivers\mouclass.sys	Kernel Driver	True	System			
	Running	OK	Normal	False	True				
mountmgr	MountMgr	c:\winnt\system32\drivers\mountmgr.sys	Kernel Driver	True	Boot	Running	OK		
	Normal	False	True						
mraid35x	mraid35x	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
mrxsmb	MRXSMB	c:\winnt\system32\drivers\mrxsmb.sys	File System Driver	True	System				
	Running	OK	Normal	False	True				
msfs	Msfs	c:\winnt\system32\drivers\msfs.sys	File System Driver	True	System				
	Running	OK	Normal	False	True				
mskssrv	Microsoft Streaming Service Proxy			c:\winnt\system32\drivers\mskssrv.sys	Kernel Driver	False			
	Manual	Stopped	OK	Normal	False	False			
mspclock	Microsoft Streaming Clock Proxy			c:\winnt\system32\drivers\mspclock.sys	Kernel Driver	False			
	Manual	Stopped	OK	Normal	False	False			
mspqm	Microsoft Streaming Quality Manager Proxy				c:\winnt\system32\drivers\mspqm.sys	Kernel Driver	False		
	Manual	Stopped	OK	Normal	False	False			
mup	Mup	c:\winnt\system32\drivers\mup.sys	File System Driver	True	Boot				
	Running	OK	Normal	False	True				
ncrc710	Ncrc710	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False
	False								
ndis	NDIS System Driver		c:\winnt\system32\drivers\ndis.sys	Kernel Driver	True	Boot			
	Running	OK	Normal	False	True				
ndistapi	Remote Access NDIS TAPI Driver			c:\winnt\system32\drivers\ndistapi.sys	Kernel Driver	True			
	Manual	Running	OK	Normal	False	True			
ndiswan	Remote Access NDIS WAN Driver			c:\winnt\system32\drivers\ndiswan.sys	Kernel Driver	True			
	Manual	Running	OK	Normal	False	True			
ndproxy	NDIS Proxy	c:\winnt\system32\drivers\ndproxy.sys	Kernel Driver	True	Manual	Running	OK		
	Normal	False	True						
netbios	NetBIOS Interface		c:\winnt\system32\drivers\netbios.sys	File System Driver	True				
	System	Running	OK	Normal	False	True			
netbt	NetBios over Tcpip		c:\winnt\system32\drivers\netbt.sys	Kernel Driver	True	System			
	Running	OK	Normal	False	True				
netdetect	NetDetect	c:\winnt\system32\drivers\netdtect.sys	Kernel Driver	False	Manual	Stopped	OK		
	Normal	False	False						
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			
openhci	Microsoft USB Open Host Controller Driver				c:\winnt\system32\drivers\openhci.sys	Kernel Driver	True		
	Manual	Running	OK	Normal	False	True			
parallel	Parallel class driver		c:\winnt\system32\drivers\parallel.sys	Kernel Driver	True	Manual			
	Running	OK	Normal	False	True				
parport	Parallel port driver		c:\winnt\system32\drivers\parport.sys	Kernel Driver	True	System			
	Running	OK	Ignore	False	True				
partmgr	PartMgr	c:\winnt\system32\drivers\partmgr.sys	Kernel Driver	True	Boot	Running	OK		
	Normal	False	True						
parvdm	ParVdm	c:\winnt\system32\drivers\parvdm.sys	Kernel Driver	True	Auto	Running	OK		
	Ignore	False	True						

APPENDIX C – TUNABLE PARAMETERS

```
pci             PCI Bus Driver                  c:\winnt\system32\drivers\pci.sys        Kernel Driver   True    Boot
                Running OK      Critical        False   True
pcidump         PCIDump         Not Available   Kernel Driver   False   System  Stopped OK      Ignore  False
                False
pciide          PCIIde          c:\winnt\system32\drivers\pciide.sys    Kernel Driver   True    Boot    Running OK
                Normal  False   True
pcmcia          Pcmcia          c:\winnt\system32\drivers\pcmcia.sys    Kernel Driver   False   Disabled        Stopped OK
                Normal  False   False
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 Mini port (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
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
ultra66         ultra66         Not Available   Kernel Driver   False   Disabled        Stopped OK      Normal  False
                False
update          Microcode Update Driver         c:\winnt\system32\drivers\update.sys    Kernel Driver   True    Manual
                Running OK      Normal  False   True
usbhub          Microsoft USB Standard Hub Driver       c:\winnt\system32\drivers\usbhub.sys    Kernel Driver   True
                Manual  Running OK      Normal  False   True
vgasave         VgaSave         c:\winnt\system32\drivers\vga.sys       Kernel Driver   True    System  Running OK
                Ignore  False   True
wanarp          Remote Access IP ARP Driver             c:\winnt\system32\drivers\wanarp.sys    Kernel Driver   True
                Manual  Running OK      Normal  False   True
wdica           WDICA           Not Available   Kernel Driver   False   Manual  Stopped OK      Ignore  False
                False
```

[Environment Variables]

```
Variable        Value           User Name
ComSpec         %SystemRoot%\system32\cmd.exe           <SYSTEM>
Os2LibPath      %SystemRoot%\system32\os2\dll;          <SYSTEM>
Path            %SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\Program Files\Microsoft SQL
Server\80\Tools\BINN    <SYSTEM>
windir          %SystemRoot%            <SYSTEM>
OS              Windows_NT <SYSTEM>
PROCESSOR_ARCHITECTURE          x86             <SYSTEM>
PROCESSOR_LEVEL 6               <SYSTEM>
PROCESSOR_IDENTIFIER            x86 Family 6 Model 8 Stepping 6, GenuineIntel           <SYSTEM>
PROCESSOR_REVISION      0806    <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>
```

APPENDIX C – TUNABLE PARAMETERS

TEMP %USERPROFILE%\Local Settings\Temp N2_CLIENT1\Administrator
TMP %USERPROFILE%\Local Settings\Temp N2_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
Unknown 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
No network connections information

[Running Tasks]

Name Path Process ID Priority Min Working Set Max Working Set Start Time
Version Size File Date
system idle process Not Available 0 0 Not Available Not Available Not Available Unknown
Unknown Unknown
system Not Available 8 8 0 1413120 Not Available Unknown Unknown
Unknown
smss.exe c:\winnt\system32\smss.exe 160 11 204800 1413120 8/8/2001 9:26:02
AM 5.00.2170.1 44.27 KB (45,328 bytes) 12/7/1999 4:00:00 AM
csrss.exe Not Available 184 13 Not Available Not Available 8/8/2001 9:26:09 AM Unknown
Unknown Unknown
winlogon.exe c:\winnt\system32\winlogon.exe 204 13 204800 1413120 8/8/2001 9:26:11
AM 5.00.2182.1 173.27 KB (177,424 bytes) 12/7/1999 4:00:00 AM
services.exe c:\winnt\system32\services.exe 232 9 204800 1413120 8/8/2001 9:26:12
AM 5.00.2134.1 86.77 KB (88,848 bytes) 12/7/1999 4:00:00 AM
lsass.exe c:\winnt\system32\lsass.exe 244 13 204800 1413120 8/8/2001 9:26:13
AM 5.00.2184.1 32.77 KB (33,552 bytes) 12/7/1999 4:00:00 AM
gnconmgr.exe c:\winnt\system32\gnconmgr.exe 384 8 204800 1413120
8/8/2001 9:26:17 AM 4.2.0.75 168.05 KB (172,079 bytes) 7/24/2001 1:58:09 PM
svchost.exe c:\winnt\system32\svchost.exe 448 8 204800 1413120 8/8/2001 9:26:18
AM 5.00.2134.1 7.77 KB (7,952 bytes) 12/7/1999 4:00:00 AM
spoolsv.exe c:\winnt\system32\spoolsv.exe 476 8 204800 1413120 8/8/2001 9:26:18
AM 5.00.2161.1 43.77 KB (44,816 bytes) 2/15/2001 6:52:41 AM
msdtc.exe c:\winnt\system32\msdtc.exe 504 8 204800 1413120 8/8/2001 9:26:19
AM 1999.9.3421.3 6.77 KB (6,928 bytes) 2/15/2001 7:05:48 AM
svchost.exe c:\winnt\system32\svchost.exe 608 8 204800 1413120 8/8/2001 9:26:21
AM 5.00.2134.1 7.77 KB (7,952 bytes) 12/7/1999 4:00:00 AM
llssrv.exe c:\winnt\system32\llssrv.exe 628 9 204800 1413120 8/8/2001 9:26:21
AM 5.00.2167.1 114.27 KB (117,008 bytes) 12/7/1999 4:00:00 AM
regsvc.exe c:\winnt\system32\regsvc.exe 668 8 204800 1413120 8/8/2001 9:26:21
AM 5.00.2155.1 65.27 KB (66,832 bytes) 12/7/1999 4:00:00 AM
mstask.exe c:\winnt\system32\mstask.exe 708 8 204800 1413120 8/8/2001 9:26:22
AM 4.71.2137.1 115.27 KB (118,032 bytes) 2/15/2001 3:11:09 PM
snmp.exe c:\winnt\system32\snmp.exe 740 8 204800 1413120 8/8/2001 9:26:23
AM 5.00.2173.1 29.77 KB (30,480 bytes) 2/15/2001 7:05:38 AM
winmgmt.exe c:\winnt\system32\wbem\winmgmt.exe 812 8 204800 1413120 8/8/2001 9:26:23
AM 1.50.1085.0001 188.05 KB (192,567 bytes) 12/7/1999 4:00:00 AM
inetinfo.exe c:\winnt\system32\inetsrv\inetinfo.exe 836 8 204800 1413120 8/8/2001 9:26:29
AM 5.00.0984 14.27 KB (14,608 bytes) 2/15/2001 7:06:21 AM
dfssvc.exe c:\winnt\system32\dfssvc.exe 896 8 204800 1413120 8/8/2001 9:26:30
AM 5.00.2191.1 85.27 KB (87,312 bytes) 12/7/1999 4:00:00 AM
explorer.exe c:\explorer.exe 1068 8 204800 1413120 8/8/2001 9:26:34 AM Not
Available 8.00 KB (8,192 bytes) 8/6/2001 5:43:11 PM
explorer.exe c:\winnt\explorer.exe 1080 8 204800 1413120 8/8/2001 9:26:35 AM
5.00.2920.0000 232.77 KB (238,352 bytes) 12/7/1999 4:00:00 AM
hpmon.exe c:\winnt\system32\hpmon.exe 1128 8 204800 1413120 8/8/2001 9:26:40
AM 1.11 28.50 KB (29,184 bytes) 2/2/2001 9:16:12 AM
acrotray.exe c:\program files\adobe\acrobat 4.0\distillr\acrotray.exe 1168 8 204800 1413120
8/8/2001 9:26:40 AM Not Available 42.50 KB (43,520 bytes) 3/6/2001 3:12:24 PM
svchost.exe c:\winnt\system32\svchost.exe 1356 8 204800 1413120 8/8/2001 9:26:53
AM 5.00.2134.1 7.77 KB (7,952 bytes) 12/7/1999 4:00:00 AM
mmc.exe c:\winnt\system32\mmc.exe 1380 8 204800 1413120 8/8/2001 9:26:54
AM 5.00.2153.1 589.27 KB (603,408 bytes) 12/7/1999 4:00:00 AM
mmc.exe c:\winnt\system32\mmc.exe 1136 8 204800 1413120 8/8/2001 9:26:59
AM 5.00.2153.1 589.27 KB (603,408 bytes) 12/7/1999 4:00:00 AM
dmremote.exe c:\winnt\system32\dmremote.exe 1072 8 204800 1413120
8/8/2001 9:27:02 AM 2191.1.296.2 10.27 KB (10,512 bytes) 12/7/1999 4:00:00 AM
dmadmin.exe c:\winnt\system32\dmadmin.exe 1344 8 204800 1413120
8/8/2001 9:27:02 AM 2191.1.296.2 144.27 KB (147,728 bytes) 12/7/1999 4:00:00 AM
rsvp.exe c:\winnt\system32\rsvp.exe 1408 8 204800 1413120 8/8/2001 9:27:40
AM 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
dmconfig.dll 2191.1.296.2 307.77 KB (315,152 bytes) 12/7/1999 4:00:00 AM VERITAS Software Corp.
c:\winnt\system32\dmconfig.dll
dmadmin.exe 2191.1.296.2 144.27 KB (147,728 bytes) 12/7/1999 4:00:00 AM VERITAS Software Corp.
c:\winnt\system32\dmadmin.exe
dmremote.exe 2191.1.296.2 10.27 KB (10,512 bytes) 12/7/1999 4:00:00 AM VERITAS Software Corp.
c:\winnt\system32\dmremote.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
dmintf.dll 2191.1.296.2 12.77 KB (13,072 bytes) 12/7/1999 4:00:00 AM VERITAS Software Corp.
c:\winnt\system32\dmintf.dll
dmdlgs.dll 2191.1.296.2 170.27 KB (174,352 bytes) 12/7/1999 4:00:00 AM Microsoft Corp., VERITAS Software
c:\winnt\system32\dmdlgs.dll
olepro32.dll 5.0.4512 160.27 KB (164,112 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\olepro32.dll
dmview.ocx 2191.1.296.2 60.27 KB (61,712 bytes) 12/7/1999 4:00:00 AM Microsoft Corp., VERITAS Software
c:\winnt\system32\dmview.ocx
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
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

APPENDIX C – TUNABLE PARAMETERS

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) 2/15/2001 3:11:16 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
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
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
acrotray.exe Not Available 42.50 KB (43,520 bytes) 3/6/2001 3:12:24 PM Not Available c:\program files\adobe\acrobat
4.0\distillr\acrotray.exe
hpmon.exe 1.11 28.50 KB (29,184 bytes) 2/2/2001 9:16:12 AM Hewlett-Packard Company
c:\winnt\system32\hpmon.exe
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
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
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
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
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
hhsetup.dll 4.74.8702 66.27 KB (67,856 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\hhsetup.dll
mmcshext.dll 5.00.2153.1 24.27 KB (24,848 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\mmcshext.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
explorer.exe Not Available 8.00 KB (8,192 bytes) 8/6/2001 5:43:11 PM Not Available c:\explorer.exe
dfssvc.exe 5.00.2191.1 85.27 KB (87,312 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\dfssvc.exe
mailmsg.dll 5.00.0984 64.27 KB (65,808 bytes) 2/15/2001 7:07:53 AM Microsoft Corporation
c:\winnt\system32\inetsrv\mailmsg.dll
ntfsdrv.dll 5.00.0984 36.77 KB (37,648 bytes) 2/15/2001 7:07:53 AM Microsoft Corporation
c:\winnt\system32\inetsrv\ntfsdrv.dll
iislog.dll 5.00.0984 76.27 KB (78,096 bytes) 2/15/2001 7:06:21 AM Microsoft Corporation
c:\winnt\system32\inetsrv\iislog.dll
ntlsapi.dll 5.00.2134.1 6.77 KB (6,928 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\ntlsapi.dll
httpext.dll 0.9.3939.9 418.27 KB (428,304 bytes) 2/15/2001 7:06:20 AM Microsoft Corporation
c:\winnt\system32\inetsrv\httpext.dll
fpexedll.dll 4.0.2.3406 20.06 KB (20,541 bytes) 2/13/2001 9:20:10 AM 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) 2/15/2001 7:06:26 AM Microsoft Corporation
c:\winnt\system32\inetsrv\md5filt.dll
aqueue.dll 5.00.0984 260.27 KB (266,512 bytes) 2/15/2001 7:07:52 AM Microsoft Corporation
c:\winnt\system32\inetsrv\aqueue.dll
gzip.dll 5.00.0984 30.27 KB (30,992 bytes) 2/15/2001 7:06:25 AM Microsoft Corporation
c:\winnt\system32\inetsrv\gzip.dll
compfilt.dll 5.00.0984 22.27 KB (22,800 bytes) 2/15/2001 7:06:25 AM Microsoft Corporation
c:\winnt\system32\inetsrv\compfilt.dll
sspifilt.dll 5.00.0984 43.27 KB (44,304 bytes) 2/15/2001 7:06:26 AM Microsoft Corporation
c:\winnt\system32\inetsrv\sspifilt.dll
seo.dll 5.00.0984 230.77 KB (236,304 bytes) 2/15/2001 7:07:53 AM Microsoft Corporation
c:\winnt\system32\inetsrv\seo.dll
iscomlog.dll 5.00.0984 24.77 KB (25,360 bytes) 2/15/2001 7:06:21 AM Microsoft Corporation
c:\winnt\system32\inetsrv\iscomlog.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
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
lonsint.dll 5.00.0984 11.77 KB (12,048 bytes) 2/15/2001 7:06:21 AM Microsoft Corporation
c:\winnt\system32\inetsrv\lonsint.dll

APPENDIX C – TUNABLE PARAMETERS

inetsloc.dll	5.00.0984	20.27 KB (20,752 bytes)	2/15/2001 7:06:22 AM	Microsoft Corporation
	c:\winnt\system32\inetsloc.dll			
w3svc.dll	5.00.0984	347.27 KB (355,600 bytes)	2/15/2001 7:06:27 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\w3svc.dll			
staxmem.dll	5.00.0984	8.27 KB (8,464 bytes)	2/15/2001 7:06:20 AM	Microsoft Corporation
	c:\winnt\system32\staxmem.dll			
exstrace.dll	5.00.0984	13.77 KB (14,096 bytes)	2/15/2001 7:06:22 AM	Microsoft Corporation
	c:\winnt\system32\exstrace.dll			
rwnh.dll	5.00.0984	10.77 KB (11,024 bytes)	2/15/2001 7:07:53 AM	Microsoft Corporation
	c:\winnt\system32\rwnh.dll			
fcachdll.dll	5.00.0984	43.77 KB (44,816 bytes)	2/15/2001 7:07:53 AM	Microsoft Corporation
	c:\winnt\system32\fcachdll.dll			
smtpsvc.dll	5.00.0984	406.27 KB (416,016 bytes)	2/15/2001 7:07:55 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\smtpsvc.dll			
iisfecnv.dll	5.00.0984	7.27 KB (7,440 bytes)	2/15/2001 7:06:21 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\iisfecnv.dll			
infocomm.dll	5.00.0984	234.27 KB (239,888 bytes)	2/15/2001 7:06:21 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\infocomm.dll			
isatq.dll	5.00.0984	61.27 KB (62,736 bytes)	2/15/2001 7:06:23 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\isatq.dll			
ftpsvc2.dll	5.00.0984	113.77 KB (116,496 bytes)	2/15/2001 7:06:27 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\ftpsvc2.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)	2/15/2001 7:06:21 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\svcext.dll			
admexs.dll	5.00.0984	27.77 KB (28,432 bytes)	2/15/2001 7:06:20 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\admexs.dll			
wamreg.dll	5.00.0984	46.27 KB (47,376 bytes)	2/15/2001 7:06:27 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\wamreg.dll			
metadata.dll	5.00.0984	70.77 KB (72,464 bytes)	2/15/2001 7:06:21 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\metadata.dll			
iismap.dll	5.00.0984	56.27 KB (57,616 bytes)	2/15/2001 7:06:22 AM	Microsoft Corporation
	c:\winnt\system32\iismap.dll			
nsepm.dll	5.00.0984	43.27 KB (44,304 bytes)	2/15/2001 7:06:21 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\nsepm.dll			
coadmin.dll	5.00.0984	39.77 KB (40,720 bytes)	2/15/2001 7:06:22 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\coadmin.dll			
iisadmin.dll	5.00.0984	14.77 KB (15,120 bytes)	2/15/2001 7:06:20 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\iisadmin.dll			
rpcref.dll	5.00.0984	4.27 KB (4,368 bytes)	2/15/2001 7:06:21 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\rpcref.dll			
inetinfo.exe	5.00.0984	14.27 KB (14,608 bytes)	2/15/2001 7:06:21 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\inetinfo.exe			
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			
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			
hpbxlk7.dll	3.38	581.50 KB (595,456 bytes)	3/6/2000 11:44:35 AM	Hewlett-Packard Corporation
	c:\winnt\system32\spool\drivers\w32x86\2\hpbxlk7.dll			
hpbxlk6.dll	3.38	380.00 KB (389,120 bytes)	3/6/2000 11:44:35 AM	Hewlett-Packard Corporation
	c:\winnt\system32\spool\drivers\w32x86\2\hpbxlk6.dll			
hpbxlk5.dll	3.38	72.00 KB (73,728 bytes)	3/6/2000 11:44:35 AM	Hewlett-Packard Corporation
	c:\winnt\system32\spool\drivers\w32x86\2\hpbxlk5.dll			
adobepsu.dll	5.01.107	152.50 KB (156,160 bytes)	3/6/2001 3:12:33 PM	Adobe Systems Incorporated
	c:\winnt\system32\spool\drivers\w32x86\2\adobepsu.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			
pdfddui.dll	4.0	159.52 KB (163,344 bytes)	3/6/2001 3:12:25 PM	Adobe Systems Inc.
	c:\winnt\system32\spool\drivers\w32x86\2\pdfddui.dll			
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)	2/15/2001 3:11:08 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			
gnisnmp.dll	Not Available	48.00 KB (49,152 bytes)	7/10/2001 11:54:25 AM	Not Available c:\winnt\system32\gnisnmp.dll
ftpmib.dll	5.00.0984	6.27 KB (6,416 bytes)	2/15/2001 7:06:27 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\ftpmib.dll			
iisrtl.dll	5.00.0984	120.77 KB (123,664 bytes)	2/15/2001 7:06:22 AM	Microsoft Corporation
	c:\winnt\system32\iisrtl.dll			
infoadmn.dll	5.00.0984	12.27 KB (12,560 bytes)	2/15/2001 7:06:22 AM	Microsoft Corporation
	c:\winnt\system32\infoadmn.dll			
httpmib.dll	5.00.0984	9.27 KB (9,488 bytes)	2/15/2001 7:06:25 AM	Microsoft Corporation
	c:\winnt\system32\inetsrv\httpmib.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			
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			
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			

APPENDIX C – TUNABLE PARAMETERS

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
evntagnt.dll 5.00.2167.1 94.77 KB (97,040 bytes) 2/15/2001 7:05:38 AM Microsoft Corporation
c:\winnt\system32\evntagnt.dll
snmpmib.dll 5.00.2134.1 5.77 KB (5,904 bytes) 2/15/2001 7:05:37 AM Microsoft Corporation
c:\winnt\system32\snmpmib.dll
hostmib.dll 5.00.2134.1 36.27 KB (37,136 bytes) 2/15/2001 7:05:37 AM Microsoft Corporation
c:\winnt\system32\hostmib.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) 2/15/2001 7:05:37 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) 2/15/2001 7:05:38 AM Microsoft Corporation
c:\winnt\system32\snmp.exe
msidle.dll 5.00.2920.0000 6.27 KB (6,416 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\msidle.dll
mstask.exe 4.71.2137.1 115.27 KB (118,032 bytes) 2/15/2001 3:11:09 PM Microsoft Corporation
c:\winnt\system32\mstask.exe
regsvc.exe 5.00.2155.1 65.27 KB (66,832 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\regsvc.exe
llsrpc.dll 5.00.2149.1 45.77 KB (46,864 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\llsrpc.dll
llssrv.exe 5.00.2167.1 114.27 KB (117,008 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\llssrv.exe
ntmsdba.dll 5.00.2187.1 167.77 KB (171,792 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\ntmsdba.dll
rasdlg.dll 5.00.2194.1 514.27 KB (526,608 bytes) 12/7/1999 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
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
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
ntmssvc.dll 5.00.2187.1 390.77 KB (400,144 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\ntmssvc.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) 2/15/2001 7:05:49 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) 2/15/2001 7:05:48 AM Microsoft Corporation
c:\winnt\system32\xolehlp.dll
msdtclog.dll 1999.9.3421.3 89.77 KB (91,920 bytes) 2/15/2001 7:05:48 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) 2/15/2001 7:05:48 AM Microsoft Corporation
c:\winnt\system32\msdtcprx.dll
txfaux.dll 1999.9.3422.24 341.27 KB (349,456 bytes) 2/15/2001 7:05:48 AM Microsoft Corporation
c:\winnt\system32\txfaux.dll
msdtctm.dll 1999.9.3422.12 1.02 MB (1,070,864 bytes) 2/15/2001 7:05:48 AM Microsoft Corporation
c:\winnt\system32\msdtctm.dll
msdtc.exe 1999.9.3421.3 6.77 KB (6,928 bytes) 2/15/2001 7:05:48 AM Microsoft Corporation
c:\winnt\system32\msdtc.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
admwprox.dll 5.00.0984 31.77 KB (32,528 bytes) 2/15/2001 7:06:22 AM Microsoft Corporation
c:\winnt\system32\admwprox.dll
inetpp.dll 5.00.2161.1 63.27 KB (64,784 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\inetpp.dll
win32spl.dll 5.00.2162.1 92.27 KB (94,480 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\win32spl.dll
usbmon.dll 5.00.2165.1 11.27 KB (11,536 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\usbmon.dll
tcpmon.dll 5.00.2165.1 40.77 KB (41,744 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\tcpmon.dll
pjlmon.dll 5.00.2165.1 12.77 KB (13,072 bytes) 11/30/1999 3:39:36 PM Microsoft Corporation
c:\winnt\system32\pjlmon.dll
adistres.dll Not Available 49.00 KB (50,176 bytes) 3/6/2001 3:12:20 PM Not Available c:\program files\adobe\acrobat 4.0\distillr\adistres.dll
pdfports.dll Not Available 16.00 KB (16,384 bytes) 3/6/2001 3:12:24 PM Not Available c:\winnt\system32\pdfports.dll
cnbjmon.dll 5.00.2134.1 43.77 KB (44,816 bytes) 11/30/1999 3:38:48 PM Microsoft Corporation
c:\winnt\system32\cnbjmon.dll
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
localspl.dll 5.00.2191.1 244.77 KB (250,640 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\localspl.dll
spoolss.dll 5.00.2161.1 61.77 KB (63,248 bytes) 2/15/2001 6:52:41 AM Microsoft Corporation
c:\winnt\system32\spoolss.dll
spoolsv.exe 5.00.2161.1 43.77 KB (44,816 bytes) 2/15/2001 6:52:41 AM Microsoft Corporation
c:\winnt\system32\spoolsv.exe
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
vipl.dll 4.2.0.75 80.00 KB (81,920 bytes) 8/2/2001 6:53:04 PM Giganet Incorporated
c:\winnt\system32\vipl.dll
gnconmgr.exe 4.2.0.75 168.05 KB (172,079 bytes) 7/24/2001 1:58:09 PM Giganet Incorporated
c:\winnt\system32\gnconmgr.exe
dssbase.dll 5.00.2150.1 140.77 KB (144,144 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\dssbase.dll
oakley.dll 5.00.2174.1 420.27 KB (430,352 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\oakley.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
polagent.dll 5.00.2183.1 108.27 KB (110,864 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\polagent.dll
scecli.dll 5.00.2191.1 105.27 KB (107,792 bytes) 12/7/1999 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

APPENDIX C – TUNABLE PARAMETERS

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
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
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
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
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
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
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
msgsvc.dll 5.00.2181.1 33.77 KB (34,576 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\msgsvc.dll
browser.dll 5.00.2142.1 48.27 KB (49,424 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\browser.dll
alrsvc.dll 5.00.2134.1 17.77 KB (18,192 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\alrsvc.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
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
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
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
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
trkwks.dll 5.00.2166.1 88.77 KB (90,896 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\trkwks.dll
trksvr.dll 5.00.2166.1 51.77 KB (53,008 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\trksvr.dll

seclogon.dll 5.00.2135.1 15.77 KB (16,144 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\seclogon.dll
psbase.dll 5.00.2146.1 111.77 KB (114,448 bytes) 12/7/1999 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
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
dnsrslvr.dll 5.00.2181.1 88.27 KB (90,384 bytes) 12/7/1999 4:00:00 AM Microsoft Corporation
c:\winnt\system32\dnsrslvr.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
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
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
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
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
clbcatq.dll 1999.9.3422.14 479.27 KB (490,768 bytes) 2/15/2001 7:05:42 AM Microsoft Corporation
c:\winnt\system32\clbcatq.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
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

APPENDIX C – TUNABLE PARAMETERS

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			
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			
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			
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	Running	Auto	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
Application Management	AppMgmt	Stopped	Manual	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
Computer Browser	Browser	Running	Auto	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
Indexing Service	cisvc	Stopped	Manual	Share Process	c:\winnt\system32\cisvc.exe			
	Normal	LocalSystem	0					
ClipBook	ClipSrv	Stopped	Manual	Own Process	c:\winnt\system32\clipsrv.exe		Normal	
	LocalSystem	0						
Distributed File System	Dfs	Running	Auto	Own Process	c:\winnt\system32\dfssvc.exe			
	Normal	LocalSystem	0					
DHCP Client	Dhcp	Running	Auto	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
Logical Disk Manager Administrative Service		dmadmin	Running	Manual	Share Process			
	c:\winnt\system32\dmadmin.exe /com	Normal	LocalSystem	0				
Logical Disk Manager	dmserver	Running	Auto	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
DNS Client	Dnscache	Running	Auto	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
Event Log	Eventlog	Running	Auto	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
COM+ Event System	EventSystem	Running	Manual	Share Process	c:\winnt\system32\svchost.exe -k			
netsvcs	Normal	LocalSystem	0					
Fax Service	Fax	Stopped	Manual	Own Process	c:\winnt\system32\faxsvc.exe		Normal	
	LocalSystem	0						
cLAN Connection Manager		GniConMgr	Running	Auto	Own Process	c:\winnt\system32\gnconmgr.exe		
	Normal	LocalSystem	0					
IIS Admin Service	IISADMIN	Running	Auto	Share Process				
	c:\winnt\system32\inetsrv\inetinfo.exe	Normal	LocalSystem	0				
Intersite Messaging	IsmServ	Stopped	Disabled	Own Process	c:\winnt\system32\ismserv.exe			
	Normal	LocalSystem	0					
Kerberos Key Distribution Center		kdc	Stopped	Disabled	Share Process			
	c:\winnt\system32\lsass.exe		Normal	LocalSystem	0			
Server	lanmanserver	Running	Auto	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					
Workstation	lanmanworkstation	Running	Auto	Share Process	c:\winnt\system32\services.exe			
	Normal	LocalSystem	0					

APPENDIX C – TUNABLE PARAMETERS

License Logging Service LicenseService Running Auto Own Process c:\winnt\system32\llssrv.exe
Normal LocalSystem 0
TCP/IP NetBIOS Helper Service LmHosts Running Auto Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Messenger Messenger Running Auto Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0
NetMeeting Remote Desktop Sharing mnmsrvc Stopped Manual Own Process c:\winnt\system32\mnmsrvc.exe
Normal LocalSystem 0
Distributed Transaction Coordinator MSDTC Running Auto Own Process c:\winnt\system32\msdtc.exe
Normal LocalSystem 0
FTP Publishing Service MSFTPSVC Running Auto 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
Network DDE NetDDE Stopped Manual Share Process c:\winnt\system32\netdde.exe
Normal LocalSystem 0
Network DDE DSDM NetDDEdsdm Stopped Manual Share Process
c:\winnt\system32\netdde.exe Normal LocalSystem 0
Net Logon Netlogon Stopped Manual Share Process c:\winnt\system32\lsass.exe
Normal LocalSystem 0
Network Connections Netman Running Manual Share Process c:\winnt\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
File Replication NtFrs Stopped Manual Own Process c:\winnt\system32\ntfrs.exe Ignore
LocalSystem 0
NT LM Security Support Provider NtLmSsp Stopped Manual Share Process
c:\winnt\system32\lsass.exe Normal LocalSystem 0
Removable Storage NtmsSvc Running Auto Share Process c:\winnt\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Plug and Play PlugPlay Running Auto Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0
IPSEC Policy Agent PolicyAgent Running Auto Share Process c:\winnt\system32\lsass.exe
Normal LocalSystem 0
Protected Storage ProtectedStorage Running Auto Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Remote Access Auto Connection Manager RasAuto Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Remote Access Connection Manager RasMan Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Routing and Remote Access RemoteAccess Stopped Disabled Share Process
c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Remote Registry Service RemoteRegistry Running Auto Own Process c:\winnt\system32\regsvc.exe
Normal LocalSystem 0
Remote Command Service RMSYS Stopped Auto Own Process c:\benchcrf\rsys.exe
Normal LocalSystem 0
Remote Procedure Call (RPC) Locator RpcLocator Stopped Manual Own Process c:\winnt\system32\locator.exe
Normal LocalSystem 0
Remote Procedure Call (RPC) RpcSs Running Auto Share Process
c:\winnt\system32\svchost -k rpcss Normal LocalSystem 0
QoS RSVP RSVP Running Manual Own Process c:\winnt\system32\rsvp.exe -s Normal
LocalSystem 0
Security Accounts Manager SamSs Running Auto Share Process
c:\winnt\system32\lsass.exe Normal LocalSystem 0
Smart Card Helper SCardDrv Stopped Manual Share Process c:\winnt\system32\scardsvr.exe
Ignore LocalSystem 0
Smart Card SCardSvr Stopped Manual Share Process c:\winnt\system32\scardsvr.exe Ignore
LocalSystem 0
Task Scheduler Schedule Running Auto Share Process c:\winnt\system32\mstask.exe
Normal LocalSystem 0
RunAs Service seclogon Running Auto Share Process c:\winnt\system32\services.exe
Ignore LocalSystem 0
System Event Notification SENS Running Auto Share Process c:\winnt\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Internet Connection Sharing SharedAccess Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs Normal LocalSystem 0
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 Running Auto Own Process c:\winnt\system32\spoolsv.exe Normal
LocalSystem 0
Performance Logs and Alerts SysmonLog Stopped Manual Own Process c:\winnt\system32\smlogsvc.exe
Normal LocalSystem 0
Telephony TapiSrv Running Manual Share Process c:\winnt\system32\svchost.exe -k tapisrv
Normal LocalSystem 0
Terminal Services TermService Stopped Disabled Own Process c:\winnt\system32\termsrv.exe
Normal LocalSystem 0
Telnet TlntSvr Stopped Manual Own Process c:\winnt\system32\tlntsvr.exe Normal
LocalSystem 0
Distributed Link Tracking Server TrkSvr Running Auto Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Distributed Link Tracking Client TrkWks Running Auto Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Uninterruptible Power Supply UPS Stopped Manual Own Process c:\winnt\system32\ups.exe
Normal LocalSystem 0
Utility Manager UtilMan Stopped Manual Own Process c:\winnt\system32\utilman.exe
Normal LocalSystem 0
Windows Time W32Time Stopped Manual Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0
World Wide Web Publishing Service W3SVC Running Auto Share Process
c:\winnt\system32\inetsrv\inetinfo.exe Normal LocalSystem 0
Windows Management Instrumentation WinMgmt Running Auto Own Process
c:\winnt\system32\wbem\winmgmt.exe Ignore LocalSystem 0
Windows Management Instrumentation Driver Extensions Wmi Running Manual Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0

[Program Groups]

Group Name Name User Name
Accessories Default User:Accessories Default User
Accessories\Accessibility Default User:Accessories\Accessibility Default User
Accessories\Entertainment Default User:Accessories\Entertainment Default User
Accessories\System Tools Default User:Accessories\System Tools Default User
Startup Default User:Startup Default User
Accessories All Users:Accessories All Users
Accessories\Accessibility All Users:Accessories\Accessibility All Users
Accessories\Communications All Users:Accessories\Communications All Users
Accessories\Entertainment All Users:Accessories\Entertainment All Users
Accessories\Games All Users:Accessories\Games All Users
Accessories\Microsoft Script Debugger All Users:Accessories\Microsoft Script Debugger All Users
Accessories\System Tools All Users:Accessories\System Tools All Users
Administrative Tools All Users:Administrative Tools All Users
Adobe Acrobat 4.0 All Users:Adobe Acrobat 4.0 All Users
Deer Hunter Demo All Users:Deer Hunter Demo All Users
GigaNet All Users:GigaNet All Users

APPENDIX C – TUNABLE PARAMETERS

Microsoft Office Tools	All Users:Microsoft Office Tools	All Users
Microsoft SQL Server	All Users:Microsoft SQL Server	All Users
Startup	All Users:Startup	All Users
WinZip	All Users:WinZip	All Users
Accessories	N2_CLIENT1\Administrator:Accessories	N2_CLIENT1\Administrator
Accessories\Accessibility	N2_CLIENT1\Administrator:Accessories\Accessibility	N2_CLIENT1\Administrator
Accessories\Entertainment	N2_CLIENT1\Administrator:Accessories\Entertainment	N2_CLIENT1\Administrator
Accessories\System Tools	N2_CLIENT1\Administrator:Accessories\System Tools	N2_CLIENT1\Administrator
Administrative Tools	N2_CLIENT1\Administrator:Administrative Tools	N2_CLIENT1\Administrator
NJStar Communicator	N2_CLIENT1\Administrator:NJStar Communicator	N2_CLIENT1\Administrator
Startup	N2_CLIENT1\Administrator:Startup	N2_CLIENT1\Administrator

[Startup Programs]

Program	Command	User Name	Location
Acrobat Assistant	c:\progra~1\adobe\acroba~1.0\distillr\acrotray.exe	All Users	Common Startup
HPmon.exe	hpmon.exe	All Users	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

[OLE Registration]

Object	Local Server
Microsoft Graph Chart	C:\PROGRA~1\MICROS~4\Office10\GRAPH.EXE
Microsoft Excel Worksheet	Not Available
Microsoft Excel Chart	Not Available
Microsoft Excel Worksheet	C:\PROGRA~1\MICROS~4\Office10\EXCEL.EXE
Microsoft Excel Chart	C:\PROGRA~1\MICROS~4\Office10\EXCEL.EXE
Microsoft Word 6.0 - 7.0 Document	Not Available
Microsoft Word 6.0 - 7.0 Picture	Not Available
Microsoft Word Document	C:\PROGRA~1\MICROS~4\Office10\WINWORD.EXE
Microsoft Word Picture	C:\PROGRA~1\MICROS~4\Office10\WINWORD.EXE
Sound (OLE2)	sndrec32.exe
Media Clip	mplay32.exe
Video Clip	mplay32.exe /avi
MIDI Sequence	mplay32.exe /mid
	Not Available
Microsoft Excel Chart	Not Available
Microsoft Excel 4.0 Macro	Not Available
Sound	Not Available
Media Clip	Not Available
	Not Available
	Not Available
Image Document	C:\PROGRA~1\WINDOW~1\ACCESS~1\ImageVue\kodakimg.exe
Microsoft PowerPoint Presentation	C:\PROGRA~1\MICROS~4\Office10\POWERPNT.EXE
Microsoft PowerPoint Slide	C:\PROGRA~1\MICROS~4\Office10\POWERPNT.EXE
WordPad Document	"%ProgramFiles%\Windows NT\Accessories\WORDPAD.EXE"
Windows Media Services DRM Storage object	Not Available
Adobe Acrobat Document	C:\PROGRA~1\Adobe\ACROBA~1.0\Acrobat\Acrobat.exe
Bitmap Image	mspaint.exe
	Not Available
	Not Available
Drawing	Not Available

[Internet Explorer 5]

[Following are sub-categories of this main category]

[Summary]

Item	Value
Version	5.00.2920.0000
Build	52920
Product ID	51876-OEM-0000957-02544
Application Path	C:\Program Files\Internet Explorer
Language	English (United States)
Active Printer	\\TPC_E60\HP LaserJet 6P/6MP - Enhanced,winspool,Ne01:
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 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
advpack.dll	5.0.2920.0	87 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
browselc.dll	5.0.2920.0	35 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
browseui.dll	5.0.2920.0	793 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
ckcnv.exe	5.0.2189.1	9 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
comctl32.dll	5.81.2920.0	540 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
crypt32.dll	5.131.2173.1	466 KB	12/7/1999 5:00:00 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 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
iexplore.exe	5.0.2920.0	59 KB	12/7/1999 5:00:00 AM	C:\Program Files\Internet Explorer	Microsoft Corporation
imagehlp.dll	5.0.2195.1	125 KB	12/7/1999 5:00:00 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 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
jobexec.dll	5.0.0.1	47 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
jscript.dll	5.1.0.4615	476 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
jsproxy.dll	5.0.2920.0	13 KB	12/7/1999 5:00:00 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 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
msjava.dll	5.0.3234.0	918 KB	12/7/1999 5:00:00 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 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
occache.dll	5.0.2920.0	86 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
ole32.dll	5.0.2181.1	966 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
oleaut32.dll	2.40.4512.1	600 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
olepro32.dll	5.0.4512.1	160 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
rsabase.dll	5.0.2150.1	129 KB	12/7/1999 5:00:00 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 5:00:00 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 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
shell32.dll	5.0.2920.0	2297 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
shlwapi.dll	5.0.2920.0	283 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
url.dll	5.0.2920.0	82 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
urlmon.dll	5.0.2920.0	427 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
vbscript.dll	5.1.0.4615	428 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
webcheck.dll	5.0.2920.0	252 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
win.com	5.0.2134.1	24 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
wininet.dll	5.0.2920.0	457 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation

APPENDIX C – TUNABLE PARAMETERS

winsock.dll	3.10.0.103	3 KB	12/7/1999 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
wintrust.dll	5.131.2143.1	162 KB	12/7/1999 5:00:00 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 5:00:00 AM	C:\WINNT\system32	Microsoft Corporation
wsock32n.dll	<File Missing>		Not Available	Not Available	Not Available	Not Available

[Connectivity]

Item	Value
Connection Preference	Never dial
EnableHttp1.1	1
ProxyHttp1.1	0

LAN Settings

AutoConfigProxy	Not Available
AutoProxyDetectMode	Enabled
AutoConfigURL	
Proxy	Enabled
ProxyServer	web-proxy.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.HEWLETT-PAJ9EYV\Local Settings\Temporary Internet Files
Total Disk Space	8667 MB
Available Disk Space	4526 MB
Maximum Cache Size	270 MB
Available Cache Size	226 MB

[List of Objects]

Program File Status	CodeBase	
Shockwave Flash Object	Installed	http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab

[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	2/15/2001 to 1/22/2101	sha1RSA

[Other People Certificates]

Issued To	Issued By	Validity	Signature Algorithm
No other people certificate information available			

[Publishers]

Name
No publisher information available

[Security]

Zone	Security Level
Local intranet	Medium-low
Trusted sites	Low
Internet	Medium
Restricted sites	High

[Office 10 Applications]

[Microsoft Word 2002]

[Summary]

Item	Value
Application name	WinWord
Version	10.
Build	2627
Product ID	Not Available
Application path	C:\Program Files\Microsoft Office\Office10\
Language	Language Neutral
System language designation	English (United States)
Active printer	\\TPC_E60\HP LaserJet 6P/6MP - Enhanced, Ne01:
Documents open	Not Available
Document name	This value is not available unless Word is running.
Document path	This value is not available unless Word is running.
Document size	Not Available
Document create date	Not Available
Templates	Not Available
Add-Ins	Not Available

[Startup Folder]

Item	Value
Startup Folder	

[COM Add-Ins]

Item	Value
Microsoft Word 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.	

[Active Document]

Item	Value

APPENDIX C – TUNABLE PARAMETERS

Microsoft Word 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Fields]

Item Value
Microsoft Word 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[File Converters]

Item Value
Microsoft Word 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Fonts]

Item Value
Microsoft Word 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Headers and Footers]

Item Value
Microsoft Word 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Hyperlinks]

Item Value
Microsoft Word 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Mail Merge]

Item Value
Microsoft Word 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Page Numbers]

Item Value
Microsoft Word 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Page Setup]

Item Value
Microsoft Word 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Styles]

Item Value
Microsoft Word 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Settings]

Item Value
Microsoft Word 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Tables]

Item Value
Microsoft Word 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Microsoft Excel 2002]

[Summary]

Item Value
Application name Excel
Version 10.
Build 2614
Product ID Not Available
Application path C:\Program Files\Microsoft Office\Office10\
Language Language Neutral

Workbooks Not Available

[COM Add-Ins]

Item Value
Microsoft Excel 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Active Workbook]

Item Value
Microsoft Excel 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Add-Ins]

Item Value
Microsoft Excel 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Charts in Active Workbook]

Item Value
Microsoft Excel 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Microsoft PowerPoint 2002]

[Summary]

Item Value
Application name POWERPNT
Version 10.
Build 2623
Product ID Not Available
Application path C:\Program Files\Microsoft Office\Office10\
Language Language Neutral

Presentations Not Available

[Active Presentation]

Item Value
Microsoft PowerPoint 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[COM Add-Ins]

APPENDIX C – TUNABLE PARAMETERS

Item Value
Microsoft PowerPoint 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Hyperlinks]

Item Value
Microsoft PowerPoint 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Microsoft Outlook 2002]

[Summary]

Item Value
Application name Not Available
Version Not Available
Build Not Available
Product ID Not Available
Language Not Available
Application Path Not Available
System Language Not Available
Mail Support Not Available
Current folder Not Available
Current item Not Available

[COM Add-Ins]

Item Value
Microsoft Outlook 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Microsoft Access 2002]

[Summary]

Item Value
Microsoft Access 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[COM Add-Ins]

Item Value
Microsoft Access 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Jet Core Components]

Item Value
Description Excel
Path C:\WINNT\System32\msexcl40.dll
Version 4.00.2927.9

Description Exchange
Path C:\WINNT\System32\msexch40.dll
Version 4.00.2927.2

Description Jet 2.x
Path C:\WINNT\System32\msrd2x40.dll
Version 4.00.2927.2

Description Jet 3.x
Path C:\WINNT\System32\msrd3x40.dll
Version 4.00.2927.4

Description Jet 4.0
Path Not Available
Version

Description Lotus
Path C:\WINNT\System32\msltus40.dll
Version 4.00.2927.9

Description ODBC
Path Not Available
Version

Description Paradox
Path C:\WINNT\System32\mspbde40.dll
Version 4.00.2927.2

Description Text
Path C:\WINNT\System32\mstext40.dll
Version 4.00.2927.11

Description Xbase
Path C:\WINNT\System32\msxbde40.dll
Version 4.00.2927.2

[Microsoft Publisher 2002]

[Summary]

Item Value
Application name Not Available
System Language Designation English (United States)
ActivePrinter \\TPC_E60\HP LaserJet 6P/6MP - Enhanced, Ne01:
Product ID Not Available
Version Not Available
Language Not Available
Registered owner Not Available
Registered company Not Available
Application path (Admin) Not Available
Application installed from Not Available
Local package Not Available
Date installed Not Available
Help link (Admin) Not Available
Product info site (Admin) Not Available
Product update site (Admin) Not Available

[Active Document]

Item Value
Microsoft Publisher 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

APPENDIX C – TUNABLE PARAMETERS

[Microsoft FrontPage 2002]

[Summary]

Item Value
Microsoft FrontPage 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Active Web]

Item Value
Microsoft FrontPage 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Active Page]

Item Value
Microsoft FrontPage 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[COM Add-Ins]

Item Value
Microsoft FrontPage 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Themes]

Item Value
Microsoft FrontPage 2002 has not been started or is not installed. Install and/or start application then press F5 to refresh.

[Microsoft Office 10 Environment]

[Last Web Connection Error]

Item Value
No error messages have been generated.

[Local Web Server Extensions]

Item Value
Server type msiis
Port number /LM/W3SVC/1:
Version 4.0.2.3406
Path C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\40

[Transport]

Item Value
Microsoft OLE DB Provider for Internet Publishing True
Microsoft Office Internet Publishing False

[ODBC Drivers]

[Core Components]

Item Value
Description Administrator
Version 3.520.7326.0
File C:\WINNT\System32\odbccp32.dll

Description Control Panel Device
Version 3.520.7326.0
File C:\WINNT\System32\odbccp32.cpl

Description Control Panel Startup
Version 3.520.7326.0
File C:\WINNT\System32\odbcad32.exe

Description Cursor Library
Version 3.520.7326.0
File C:\WINNT\System32\odbccr32.dll

Description Driver Manager
Version 3.520.7326.0
File C:\WINNT\System32\odbc32.dll

Description Localized Resource DLL
Version 3.520.7326.0
File C:\WINNT\System32\odbcint.dll

Description Unicode Cursor Library
Version 3.520.7326.0
File C:\WINNT\System32\odbccu32.dll

[Drivers]

Item Value
Name SQL Server
File SQLSRV32.dll
Date 4/11/2001 3:28:48 PM
Version 2000.80.380.0

Name Microsoft Access Driver (*.mdb)
File odbcjt32.dll
Date 12/7/1999 5:00:00 AM
Version 4.0.4403.2

Name Microsoft Text Driver (*.txt; *.csv)
File odbcjt32.dll
Date 12/7/1999 5:00:00 AM
Version 4.0.4403.2

Name Microsoft Excel Driver (*.xls)
File odbcjt32.dll
Date 12/7/1999 5:00:00 AM
Version 4.0.4403.2

Name Microsoft dBase Driver (*.dbf)
File odbcjt32.dll
Date 12/7/1999 5:00:00 AM

APPENDIX C – TUNABLE PARAMETERS

Version 4.0.4403.2

Name Microsoft Paradox Driver (*.db)
File odbcjt32.dll
Date 12/7/1999 5:00:00 AM
Version 4.0.4403.2

Name Microsoft Visual FoxPro Driver
File vfpodbc.dll
Date 12/7/1999 5:00:00 AM
Version 6.1.8629.1

Name Microsoft FoxPro VFP Driver (*.dbf)
File vfpodbc.dll
Date 12/7/1999 5:00:00 AM
Version 6.1.8629.1

Name Microsoft dBase VFP Driver (*.dbf)
File vfpodbc.dll
Date 12/7/1999 5:00:00 AM
Version 6.1.8629.1

Name Microsoft Access-Treiber (*.mdb)
File odbcjt32.dll
Date 12/7/1999 5:00:00 AM
Version 4.0.4403.2

Name Microsoft Text-Treiber (*.txt; *.csv)
File odbcjt32.dll
Date 12/7/1999 5:00:00 AM
Version 4.0.4403.2

Name Microsoft Excel-Treiber (*.xls)
File odbcjt32.dll
Date 12/7/1999 5:00:00 AM
Version 4.0.4403.2

Name Microsoft dBase-Treiber (*.dbf)
File odbcjt32.dll
Date 12/7/1999 5:00:00 AM
Version 4.0.4403.2

Name Microsoft Paradox-Treiber (*.db)
File odbcjt32.dll
Date 12/7/1999 5:00:00 AM
Version 4.0.4403.2

Name Microsoft Visual FoxPro-Treiber
File vfpodbc.dll
Date 12/7/1999 5:00:00 AM
Version 6.1.8629.1

Name Driver do Microsoft Access (*.mdb)
File odbcjt32.dll
Date 12/7/1999 5:00:00 AM
Version 4.0.4403.2

Name Driver da Microsoft para arquivos texto (*.txt; *.csv)
File odbcjt32.dll
Date 12/7/1999 5:00:00 AM
Version 4.0.4403.2

Name Driver do Microsoft Excel(*.xls)
File odbcjt32.dll
Date 12/7/1999 5:00:00 AM
Version 4.0.4403.2

Name Driver do Microsoft dBase (*.dbf)
File odbcjt32.dll
Date 12/7/1999 5:00:00 AM
Version 4.0.4403.2

Name Driver do Microsoft Paradox (*.db)
File odbcjt32.dll
Date 12/7/1999 5:00:00 AM
Version 4.0.4403.2

Name Driver para o Microsoft Visual FoxPro
File vfpodbc.dll
Date 12/7/1999 5:00:00 AM
Version 6.1.8629.1

Name Microsoft ODBC for Oracle
File msorcl32.dll
Date 3/27/2001 12:13:52 PM
Version 2.573.7326.0

[OLE DB Providers]

Item	Value
MediaCatalogDB OLE DB Provider	Not Available
Microsoft OLE DB Provider for SQL Server	2000.80.380.0
Microsoft OLE DB Provider for DTS Packages	2000.80.382.0
MediaCatalogMergedDB OLE DB Provider	Not Available
Microsoft OLE DB Provider For Data Mining Services	8.0.3.23
MSDataShape	2.61.7326.0
OLE DB Provider for Microsoft Directory Services	5.0.2172.1
MediaCatalogWebDB OLE DB Provider	Not Available
Microsoft OLE DB DM Enumerator	8.0.3.23
Microsoft OLE DB Provider for Olap Services 8.0	8.0.384.0
Microsoft OLE DB Provider for Olap Services 8.0	8.0.384.0
Microsoft OLE DB Provider for Internet Publishing	10.145.3812.0
Microsoft OLE DB Provider for ODBC Drivers	2.61.7326.0
Microsoft OLE DB Enumerator for ODBC Drivers	2.61.7326.0
Microsoft Jet 4.0 OLE DB Provider	4.0.2927.2
Microsoft OLE DB Enumerator for SQL Server	2000.80.380.0
Microsoft OLE DB Simple Provider	2.61.7326.0
Microsoft OLE DB Provider for Oracle	2.61.7326.0
Microsoft OLE DB Provider for Indexing Service	5.0.2135.1

[Jet Core Components]

Item	Value
Description	Excel
Path	C:\WINNT\System32\msexcl40.dll

APPENDIX C – TUNABLE PARAMETERS

Version 4.00.2927.9

Description Exchange
Path C:\WINNT\System32\msexch40.dll
Version 4.00.2927.2

Description Jet 2.x
Path C:\WINNT\System32\msrd2x40.dll
Version 4.00.2927.2

Description Jet 3.x
Path C:\WINNT\System32\msrd3x40.dll
Version 4.00.2927.4

Description Jet 4.0
Path Not Available
Version

Description Lotus
Path C:\WINNT\System32\msltus40.dll
Version 4.00.2927.9

Description ODBC
Path Not Available
Version

Description Paradox
Path C:\WINNT\System32\mspbde40.dll
Version 4.00.2927.2

Description Text
Path C:\WINNT\System32\mstext40.dll
Version 4.00.2927.11

Description Xbase
Path C:\WINNT\System32\msxbde40.dll
Version 4.00.2927.2

[ADO Core Components]

Item Value
Description ADOConnectObject.ADOConnectObject.2
File Unknown
Version 7.00.9122.0

Description ADODB.Command
File Unknown
Version 2.61.7326.0

Description ADODB.Command
File Unknown
Version 2.61.7326.0

Description ADOMD.Catalog
File Unknown
Version 2.61.7326.0

Description ADOR.Recordset
File Unknown
Version 2.61.7326.0

Description ADOX.Catalog
File Unknown
Version 2.61.7326.0

Description RDS.DataControl
File Unknown
Version 2.61.7326.0

Description JRO.JetEngine
File Unknown
Version 2.61.7326.0

[RDO Core Components]

Item Value
Description MicrosoftRDO.rdoEngine2.0
File Unknown
Version 6.00.8862

Description MicrosoftRDO.rdoQuery2.0
File Unknown
Version 6.00.8862

Description MicrosoftRDO.rdoConnection2.0
File Unknown
Version 6.00.8862

[Office Event/Application Fault]

Item Value
No Office Event/Application Fault Entries in the Event Log

C.10 RTE Input Parameters

Profile: profile_11rte_3cl_3036
File Path: C:\benchcrf\profile_11rte_3cl_3036.pro
Version: 1.0.1

Number of Engines: 33

Name: DRIVER1
Description: n2_rte1a
Directory: c:\log\n2_rte11.log
Machine: n2_rte1
Parameter Set: perfmix
Index: 8888
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER36673265
Connect Rate: 250
Start Rate: 0

APPENDIX C – TUNABLE PARAMETERS

CLIENT_NURAND: 208
CPU: 0

Name: DRIVER10
Description: n2_rte10a
Directory: c:\log\n2_rte101.log
Machine: n2_rte10
Parameter Set: perfmix
Index: 900000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER10787060078
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER11
Description: n2_rte11a
Directory: c:\log\n2_rte111.log
Machine: n2_rte11
Parameter Set: perfmix
Index: 1000000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER11787120140
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER12
Description: n2_rte1b
Directory: c:\log\n2_rte12.log
Machine: n2_rte1
Parameter Set: perfmix
Index: 1100000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER12787241375
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER13
Description: n2_rte2b
Directory: c:\log\n2_rte22.log
Machine: n2_rte2
Parameter Set: perfmix
Index: 1200000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER13789192203
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER14
Description: n2_rte3b
Directory: c:\log\n2_rte32.log
Machine: n2_rte3
Parameter Set: perfmix
Index: 1300000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER14789227640
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER15
Description: n2_rte4b
Directory: c:\log\n2_rte42.log
Machine: n2_rte4
Parameter Set: perfmix
Index: 1400000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER15789246781
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER16
Description: n2_rte5b
Directory: c:\log\n2_rte52.log
Machine: n2_rte5
Parameter Set: perfmix
Index: 1500000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER16789266031
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER17
Description: n2_rte6b
Directory: c:\log\n2_rte62.log
Machine: n2_rte6
Parameter Set: perfmix
Index: 1600000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER17789415031
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER18

APPENDIX C – TUNABLE PARAMETERS

Description: n2_rte7b
Directory: c:\log\n2_rte72.log
Machine: n2_rte7
Parameter Set: perfmix
Index: 1700000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER18789438046
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER19
Description: n2_rte8b
Directory: c:\log\n2_rte82.log
Machine: n2_rte8
Parameter Set: perfmix
Index: 1800000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER19789459968
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER2
Description: n2_rte2a
Directory: c:\log\n2_rte21.log
Machine: n2_rte2
Parameter Set: perfmix
Index: 100000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER21082250
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER20
Description: n2_rte9b
Directory: c:\log\n2_rte92.log
Machine: n2_rte9
Parameter Set: perfmix
Index: 1900000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER20789478296
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER21
Description: n2_rte10b
Directory: c:\log\n2_rte102.log
Machine: n2_rte10
Parameter Set: perfmix
Index: 2000000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER21789643687
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER22
Description: n2_rte11b
Directory: c:\log\n2_rte112.log
Machine: n2_rte11
Parameter Set: perfmix
Index: 2100000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER22789671125
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER23
Description: n2_rte1c
Directory: c:\log\n2_rte13.log
Machine: n2_rte1
Parameter Set: perfmix
Index: 2094967296
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER23789687500
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER24
Description: n2_rte2c
Directory: c:\log\n2_rte23.log
Machine: n2_rte2
Parameter Set: perfmix
Index: 488251651
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER24789723296
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER25
Description: n2_rte3c
Directory: c:\log\n2_rte33.log
Machine: n2_rte3
Parameter Set: perfmix

APPENDIX C – TUNABLE PARAMETERS

Index: 47832909
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER25789999890
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER26
Description: n2_rte4c
Directory: c:\log\n2_rte43.log
Machine: n2_rte4
Parameter Set: perfmix
Index: 758920374
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER26790032484
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER27
Description: n2_rte5c
Directory: c:\log\n2_rte53.log
Machine: n2_rte5
Parameter Set: perfmix
Index: 748921374
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER27790051796
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER28
Description: n2_rte6c
Directory: c:\log\n2_rte63.log
Machine: n2_rte6
Parameter Set: perfmix
Index: 74239847
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER28790071328
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER29
Description: n2_rte7c
Directory: c:\log\n2_rte73.log
Machine: n2_rte7
Parameter Set: perfmix
Index: -570980864
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER29864649359
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER3
Description: n2_rte3a
Directory: c:\log\n2_rte31.log
Machine: n2_rte3
Parameter Set: perfmix
Index: 200000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER323492218
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER30
Description: n2_rte8c
Directory: c:\log\n2_rte83.log
Machine: n2_rte8
Parameter Set: perfmix
Index: 2874289
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER30864696921
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER31
Description: n2_rte9c
Directory: c:\log\n2_rte93.log
Machine: n2_rte9
Parameter Set: perfmix
Index: 841262419
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER31864716562
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER32
Description: n2_rte10c
Directory: c:\log\n2_rte103.log
Machine: n2_rte10
Parameter Set: perfmix
Index: 94717421
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER32864739828

APPENDIX C – TUNABLE PARAMETERS

Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER33
Description: n2_rte11c
Directory: c:\log\n2_rte113.log
Machine: n2_rte11
Parameter Set: perfmix
Index: 47823943
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER33158366359
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER4
Description: n2_rte4a
Directory: c:\log\n2_rte41.log
Machine: n2_rte4
Parameter Set: perfmix
Index: 300000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER423618718
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER5
Description: n2_rte5a
Directory: c:\log\n2_rte51.log
Machine: n2_rte5
Parameter Set: perfmix
Index: 400000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER5783108109
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER6
Description: n2_rte6a
Directory: c:\log\n2_rte61.log
Machine: n2_rte6
Parameter Set: perfmix
Index: 500000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER6783139421
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER7
Description: n2_rte7a
Directory: c:\log\n2_rte71.log
Machine: n2_rte7
Parameter Set: perfmix
Index: 600000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER7783159265
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER8
Description: n2_rte8a
Directory: c:\log\n2_rte81.log
Machine: n2_rte8
Parameter Set: perfmix
Index: 700000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER8783180484
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER9
Description: n2_rte9a
Directory: c:\log\n2_rte91.log
Machine: n2_rte9
Parameter Set: perfmix
Index: 800000000
Seed: 28814
Configured Users: 920
Pipe Name: DRIVER9786962593
Connect Rate: 250
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Number of User groups: 33

Driver Engine: DRIVER1
IIS Server: n2_clienta
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 1 - 92
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

APPENDIX C – TUNABLE PARAMETERS

Driver Engine: DRIVER31
IIS Server: n2_clientc
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 2761 - 2852
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER32
IIS Server: n2_clientc
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 2853 - 2944
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER18
IIS Server: n2_clientb
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 1565 - 1656
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER33
IIS Server: n2_clientc
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 2945 - 3036
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER4
IIS Server: n2_clienta
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 277 - 368
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER5
IIS Server: n2_clienta
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 369 - 460
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER11
IIS Server: n2_clienta
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 921 - 1012
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER6
IIS Server: n2_clienta
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 461 - 552
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER7
IIS Server: n2_clienta
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 553 - 644
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER8
IIS Server: n2_clienta
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 645 - 736
w_id Max Warehouse: 3036

APPENDIX C – TUNABLE PARAMETERS

Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER25
IIS Server: n2_clientc
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 2209 - 2300
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER9
IIS Server: n2_clienta
SQL Server: prf_sut6
User: sa
Protocol: Html
w_id Range: 737 - 828
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER12
IIS Server: n2_clientb
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 1013 - 1104
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER13
IIS Server: n2_clientb
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 1105 - 1196
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER19
IIS Server: n2_clientb
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 1657 - 1748
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER14
IIS Server: n2_clientb
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 1197 - 1288
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER15
IIS Server: n2_clientb
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 1289 - 1380
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER16
IIS Server: n2_clientb
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 1381 - 1472
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER17
IIS Server: n2_clientb
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 1473 - 1564
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER2
IIS Server: n2_clienta
SQL Server: prf_sut5

APPENDIX C – TUNABLE PARAMETERS

User: sa
Protocol: Html
w_id Range: 93 - 184
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER20
IIS Server: n2_clientb
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 1749 - 1840
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER26
IIS Server: n2_clientb
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 2301 - 2392
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER21
IIS Server: n2_clientb
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 1841 - 1932
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER22
IIS Server: n2_clientb
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 1933 - 2024
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER23
IIS Server: n2_clientc
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 2025 - 2116
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER24
IIS Server: n2_clientc
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 2117 - 2208
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER10
IIS Server: n2_clienta
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 829 - 920
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER27
IIS Server: n2_clientc
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 2393 - 2484
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER28
IIS Server: n2_clientc
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 2485 - 2576
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

APPENDIX C – TUNABLE PARAMETERS

Driver Engine: DRIVER29
IIS Server: n2_clientc
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 2577 - 2668
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER3
IIS Server: n2_clienta
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 185 - 276
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Driver Engine: DRIVER30
IIS Server: n2_clientc
SQL Server: prf_sut5
User: sa
Protocol: Html
w_id Range: 2669 - 2760
w_id Max Warehouse: 3036
Scale: Normal
User Count: 920
District id: 1
Scale Down: No

Number of Parameter Sets: 3

perf-mix
New Parameter Set

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	44.89		12.05	18.01	0.10	5.00	0.10
Payment	43.02		12.05	3.01	0.10	5.00	0.10
Delivery	4.02		5.05	2.01	0.10	5.00	0.10
Stock Level	4.03		5.05	2.01	0.10	20.00	0.10
Order Status	4.04		10.05	2.01	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

perfmix
New Parameter Set

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	44.90		12.05	18.01	0.10	5.00	0.10
Payment	43.01		12.05	3.01	0.10	5.00	0.10
Delivery	4.02		5.05	2.01	0.10	5.00	0.10
Stock Level	4.03		5.05	2.01	0.10	20.00	0.10
Order Status	4.04		10.05	2.01	0.10	5.00	0.10

Appendix D Disk Storage

TPC-C 60 Day Space Requirements						
Warehouses	**3036**				**TpmC**	**37,596.34**
Table	**Rows**	**Data KB**	**Index KB**	**Extra 5% KB**	**8hr Space**	**Total Space KB**
Warehouse	**3036**	**328**	**32**	18		378
District	**30360**	**3,376**	**32**	170		3578
Customer	**91080000**	**66240008**	**3949912**	3,509,496		73699416
History	**91080000**	**5060008**	**96**		1,013,410	5060104
NewOrder	**27324000**	**432008**	**1,016**			433024
Orders	**91080000**	**2791728**	**1269504**		1,301,210	4061232
OrderLine	**910798313**	**56924896**	**120504**		11,283,213	57045400
Item	**100000**	**9528**	**48**	479		10055
Stock	**303600000**	**97152008**	**181608**	4,866,681		102200297
Total		228,613,888	5,522,752	8,376,844	13,597,832	242,513,484

	MB	
Dynamic Space	63,258	Sum of Data for Order, Orderline and History
Static Space	173,571	Sum of Data+Index+5%-Dynamic Space
Free Space	na	Total Allocated Spac - (Dynamic + Static Space)
Daily Growth	12,534	(Dynamic Space/(W*62.5))*tpmc
Daily Spread	-	(Free Space -1.5*Daily Growth) Zero Assumed

60 Day Space MB	925,598					
60 Day Space GB	**903.90**	**GB**				
Log Size	101,400	MB				
KB Per New Order	5.0608	KB				
8 hr log MB	89,188	MB				
8 hr log GB	**87.0975**	**GB**				
Space Usage	**GB Needed**	**Disks Measured**	**Disk Size**	**Formatted Size**		
60 Day Space DB	903.90	144	18GB	17.35		
		0	9GB	8.460		
		0	4GB	4.00		
Total DB		144.00		2,498.40	**OK**	
8-hr log + mirror	174.1950	12	18GB	17.35	**OK**	208.20
OS, Swap	3	1	9GB	8.460		
Total Storage Needed	**1,081.10**	**GB**	**Configured:**	**2,515.75**		

APPENDIX D – DISK STORAGE

tpmC	37,596.34							
	Data Before KB	Index Before KB	Data After KB	Index After KB	Data Grow KB	KB/New-Order	8-Hr Growth KB	8-Hr Growth MB
History	5,134,680	120	5,508,216	144	373,536	0.0562	1,013,409.59	989.66
Order	3,034,600	2,539,080	3,505,320	2,543,544	475,184	0.0721	1,301,209.66	1,270.71
Order-Line	58,163,288	240,960	62,322,448	240,984	4,159,160	0.6252	11,283,212.71	11,018.76
								13,279.13
	sum(*) Before		sum(*) After		Num New-			
d_next_o_id	92,420,866		99,073,018		6,652,152			
	Before MB		After MB		Grow MB	KB/New-Order	8-Hr Growth MB	8-Hr Growth GB
Log	9182.074507		42058.23652		32876.16	5.0608	89,187.86	87.10
						5,182.2560		2.50
101399.9922	9.055301		41.477554					
Database tpcc log used (%)								

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 LH6000 August 15th, 2001				
Description	**Part #**	**Qty**	**Price**	**Extended**
Server				
HP Netserver LH 6000 - Six Pentium III Xeon 700MHz 2MB L2 cache, tower unit. Includes: 0 MB PC 133 SDRAM, integrated dual channel HP NetRAID controller, integrated 10/100TX NIC, CD ROM, flexible disk drive, hot swap power supplies and fans, integrated HP Remote assistant	P4583AV	1	$18,960	$18,960
512 MB DIMM kit	D8267AV	16	$610	$9,760
HP 15" Color Monitor	D8896A	1	$144	$144
HP NetServer mini-DIN keyboard and mouse	D4950B/C3751B	1	$79	$79
HP 9.1 GB 10K HotSwap Wide Ultra2 SCSI Disk	P1168A	1	$262	$262
HP Rack System/E41 (41 Units usable space)	J1500A	1	$1,590	$1,590
HP Power Distribution Unit 120-240V	E5929A	2	$234	$468
HP SureStore DAT24i Internal Tape Drive	C1555D	1	$759	$759
APC Smart-UPS 3000	588293	3	$1,340	$4,020
HP LH 6000 Support Pack 3 years, 24x7, 4 hr response	H3127A	1	$1,715	$1,715
Storage				
HP Netserver Rack Storage/12	D5989B	13	$1,890	$24,570
HP RS/12 Support Pack 3 years, 24x7, 4 hr response	H5513A	13	$1,450	$18,850
HP 18GB, 15krpm Hot-swap disk module	D9421A	156	$669	$104,364
Mylex eXtremeRAID 2000	E2000-4-32MB	6	$1,872	$11,232
HP SCSI Cable 2.5m UDHTS 68/HDTS 68	D6020A	13	$84	$1,092
Clients				
HP Netserver E800 Pentium III 866MHz	P2457A	3	$832	$2,496
128MB DRAMs for E60	D8265A	12	$157	$1,884
HP 9.1 GB 10K Ultra3 SCSI Disk	P1214A	3	$167	$501
HP 15" Color Monitor	D8896A	3	$144	$432
HP E800 Support Pack 3 years, 24x7, 4 hr response	H4636E	3	$745	$2,235
HP Procurve Switch 2424M	J4093A	2	$861	$1,722
Software				
Microsoft SQL Server 2000 Ent Ed. Per Processor license	810-00846	6	$14,869	$89,214
Microsoft Windows 2000 Advanced Server	C10-00475	1	$2,160	$2,160
Microsoft Windows 2000 Server	C11-00016	3	$839	$2,517
Microsoft Visual C++ Professional 6.0	716856	1	$449	$449
		Total		**$301,475**

APPENDIX E – PRICE QUOTATIONS

580 Main Street • Bolton, MA 01740 • 888-228-3342
www.emulex.com

Quote Number: 010814-01
Date: August 14, 2001
Issued by: Robert G. Hamilton
E-mail: rob.hamilton@emulex.com
Phone: (978) 779-7281
Fax: (978) 779-7203

Presented to: Mr. Alexander Carlton
Hewlett-Packard
Netserver Performance Team
Phone: n/a
Fax: (408) 447-8001
Email: alex_carlton@hp.com

Description: cLAN provides full bandwidth connections between all nodes, allowing 2.5 Gb/s full duplex communication between all nodes simultaneously, i.e. each node has a 1.25 Gb/s pipe available to any other node. Each server uses one Host Adapter.

Quantity	Product Description	Unit Price	Extended
6	**cLAN1000** 32/64Bit PCI Host Adapter with copper interface *Software (cLAN-S002) provided at no charge, at least one copy required*	$795.00	$4,770.00
3	**cLAN5000** Thirty (30) port Cluster Switch with copper interface	$6,250.00	$18,750.00
6	**cLAN-A0211** 2 Meter Copper Cable, *Switch Interconnects*	$95.00	$570.00
1	**cLAN-S002** cLAN for Windows Software. *Includes VI driver ; TCP/IP driver ; cLANview management tool; Giganet Management Console ; NDIS*	$25.00	$0.00
	Actual shipping charges will be billed.	**TOTAL:**	**$24,090.00**

F.O.B.	Terms
Bolton, MA	Net 30

APPENDIX E – PRICE QUOTATIONS

Warranty Statement

Hardware: Emulex warrants that each of its products will perform in accordance with its specifications and shall be free from defects in material and workmanship under normal use for a period of time from date of purchase as follows:

Fibre Channel
LPxxxx Host Adapters two years *(1)
LHxxxx Hubs one year **(2)

*(1) The Warranty period will automatically be extended to 3 years for products that are registered within 90 days of purchase. (Applies to product registered after January 1, 1997).
**(2) These models include a one-year complimentary Advanced Replacement Program subscription, when registered within 90 days of purchase (North American Only).

Media: Emulex warrants all media on which software is furnished to be free of defects in material for a period of thirty (30) days from the date of product purchase.

Cable and Kit Accessories: All Emulex provided cables, GLM's, GBIC's, loopbacks, couplers, and kit accessories are warranted for ninety (90) days from date of purchase.

Return to Factory: During the warranty period, if the customer experiences difficulties with an Emulex product and is unable to resolve the problem via the phone with Emulex's Technical Support, a Return Authorization will be issued. Following receipt of a Return Authorization, the customer is responsible for returning the product to Emulex, freight prepaid. Emulex, upon verification of warranty, will, at its option, repair or replace the product in question and return it to the customer, freight prepaid.

Onsite services are not included as a part of these warranties.

General Terms: The above warranties shall not apply to expendable components such as fuses, bulbs, and the like, nor to connectors and other items not a part of the basic product. Nor does the foregoing warranty extend to experimental or developmental products, or to products which have been subject to misuse, neglect, accident, improper media usage or abuse; have been improperly wired, repaired, or altered by anyone other than Emulex or an authorized repair agent; have been used in material violation of Emulex's operating instructions; or have had their serial numbers removed, defaced or altered. Emulex shall have no obligation to make repairs or to cause replacement required through normal wear and tear or necessitated in whole or in part by catastrophe, fault, or negligence of the user, improper or unauthorized use of the product, or use of the product in such a manner for which it was not designed, or by causes external to the product, such as, but not limited to, power failure or air conditioning. Emulex's sole obligation hereunder shall be to repair or replace items covered in the above warranties. Customer shall provide for removal of the defective product, shipping charges for return to Emulex, and installation of its replacement. Above warranties are subject to change without notice.

Product End of Life: For Emulex products that are discontinued, notification and the date of discontinuance may be found on the Emulex web site at 'www.emulex.com/ss/discontinue.html'. Return to Factory qualified repair service shall be available for such products for seven years from such date of discontinuance.

Returned Products: Warranty claims must be received by Emulex within the applicable warranty period. All returned material must be accompanied by a Return Authorization Number assigned by Emulex.

In the event product(s) returned under the provisions of this Warranty are subsequently determined by Emulex to be functionally operational and in accordance with its published specifications, i.e., "No Defect Found" (NDF), Purchaser will be charged an NDF fee and the product shall be returned to the Purchaser freight collect.

THE EXPRESSED WARRANTIES SET FORTH IN THIS AGREEMENT ARE IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, AND ALL SUCH OTHER WARRANTIES ARE HEREBY DISCLAIMED AND EXCLUDED BY EMULEX. THE STANDARD EXPRESSED WARRANTIES ARE IN LIEU OF ALL OBLIGATIONS OR LIABILITIES ON THE PART OF EMULEX FOR DAMAGES, INCLUDING, BUT NOT LIMITED TO, SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE PRODUCTS.

Customer's sole and exclusive remedy for any breach of this warranty, whether in contract, tort, strict liability, or otherwise, shall be Emulex's obligation to repair or replace as set forth above.