UNISYS

TPC Benchmark™ C Full Disclosure Report

Unisys Corporation
Enterprise Systems

Aquanta ES5085R Server (4P)

using

Microsoft SQL Server 7.0, Enterprise Edition

on

Microsoft NT Server 4.0, Enterprise Edition

First Edition
October 26th 1999

Unisys Part Number 4500 5113-000

First Edition – October 1999

Unisys Corporation 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. Unisys Corporation 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, Unisys Corporation and Microsoft Corporation provide no warranty on the pricing information in this document.

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 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, and therefore results obtained in other operating environments may vary significantly. Unisys Corporation and Microsoft Corporation do 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 USA, October 1999.

Unisys Corporation Part Number: 4500 5113-000

Page Status

Page	Issue
i through xii	-000
0-1 through 0-3	-000
0-4	Blank
1-1 through 1-1	-000
1-2	Blank
2-1 through 2-2	-000
3-1 through 3-4	-000
4-1 through 4-7	-000
4-8	Blank
5-1 through 5-8	-000
6-1 through 6-2	-000
7-1 through 7-2	-000
8-1 through 8-1	-000
8-2	Blank
9-1 through 9-3	-000
9-4	Blank
A-1 through A-41	-000
A-42	Blank
B-1 through B-50	-000
C-1 through C-73	-000
C-74	Blank
D-1 through D-3	-000
D-4	Blank
E-1 through E-2	-000
F-1 through F-5	-000

Unisys uses an 11-digit document numbering system. The suffix of the document number (1234 5678-xyz) indicates the document level. The first digit of the suffix (x) designates a revision level; the second digit (y) designates an update level. For example, the first release of a document has a suffix of -000. A suffix of -130 designates the third update to revision 1. The third digit (z) is used to indicate an errata for a particular level and is not reflected in the page status summary.

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark C (TPC-C) conducted on the Unisys Corporation Aquanta ES5085R Server (4P). The operating system on the server was Microsoft Windows NT Server 4.0, Enterprise Edition. The DBMS used was Microsoft SQL Server 7.0, Enterprise Edition. The operating system on the clients was Microsoft Windows 2000 Server. The clients ran Microsoft's Internet Information Server 5.0 and COM+.

TPC Benchmark Metrics

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

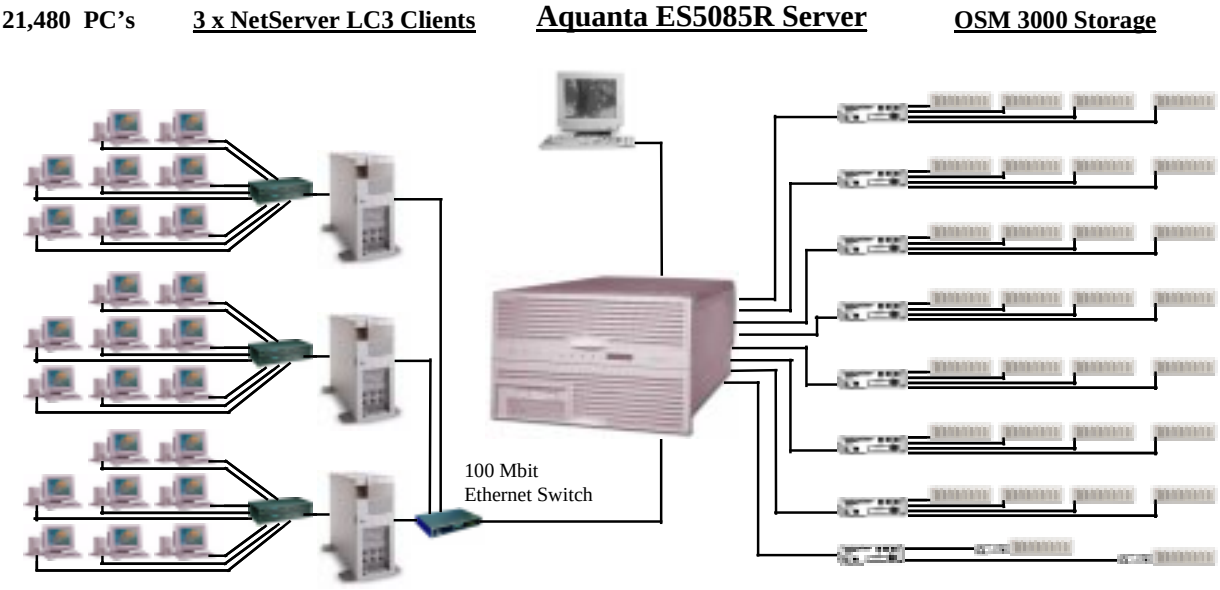
Executive Summary

The following pages contain the executive summary results of the benchmark.

Auditor

The benchmark configuration, environment, and methodology used to produce and validate the test results, along with the pricing model used to calculate the cost per tpmC, were audited by Tom Sawyer of Performance Metrics, Inc. to verify compliance with the relevant TPC specification.

UNISYS	Aquanta ES5085R Server (4P) (4P 550MHz/2MB)			TPC-C Rev. 3.5 Report Date: 26-Oct-1999
Total System Cost	TPC-C Throughput	Price/Performance		Availability Date
$468,169	**26,922.60 tpmC**	**$17.39 per tpmC**		**31-Dec-1999 ***
Processors	Database Manager	Operating System	Other Software	Number of Users
4 Pentium® III Xeon 550 MHz 2MB L2 cache	Microsoft SQL Server Enterprise Edition 7.0	Microsoft NT Server 4.0 Enterprise Edition	Windows 2000 Server, IIS 5.0 and COM+	**21,480**



System Components	Server		Clients	
	Quantity	Type	Quantity	Type
Processors	4	550 MHz Pentium® III Xeon with 2MB Level 2 Cache	3	1 x 550MHz Pentium® III with 512KB Level 2 Cache
Memory	1	4096MB	3	384MB
Disk Controllers	8 + 2 1	SCSI RAID Inbuilt SCSI	3	Inbuilt SCSI
Disk Drives	1 240	8.50 GB (boot drive) 8.54 GB	3	3.97 GB
Total Storage		2058.58 GB		11.91 GB
CD-ROM / Tape	1	SCSI CD-ROM Drive	3	CD-ROM Drive

* All hardware components are available now.

UNISYS	Aquanta ES5085R Server (4P) (4P 550MHz/2MB)						TPC-C Rev 3.5 26-Oct-1999
Description	**Style**	**Third Party Brand**	**Pricing**	**Unit Price**	**Qty.**	**Extended Price**	**5 Years Maint.**
Server Hardware							
SYS: Aquanta ES5085R, w/ CDRom, 0 Proc, 0MB Mem	ESR508151-GZN		1	$18,196	1	$18,196	$5,136
PROC: 550MHz Pentium III Xeon /2MB Cache & VRMs	XEO3550-2MB		1	$5,893	4	$23,572	$6,624
BRD: Processor Mezzanine Board, 0 Proc.	ESR81-MEZ		1	$1,179	1	$1,179	$312
MEM: 128 MB Memory, SDRAM, Buf 6ns	DIM6168-128		1	$442	32	$14,144	$7,680
BRD: Memory Carrier Board, 0 Mem.	ESR81-MCB		1	$737	1	$737	$192
MEM: Cache Coherency Filter, 4x SRAM	ESR81-CC4		1	$921	2	$1,842	$480
DISK: 9GB, 10K SCSI LVD, SCA	HDL91102-CX1		1	$516	1	$516	$240
ETHERNET: 10/100Mbit/sec, PCI 32-bit	ETH1010052-PCI		1	$100	1	$100	
SYS MGT: ES5080 Value Add Software	ESS508011-N		1	$368	1	$368	$144
MONITOR: 15-inch Color	EVG2100-P		1	$221	1	$221	
KEYBD: 104 Key Spacesaver	PCK104-SKB		1	$26	1	$26	
MOUSE: 2 Button PS2	PWM1-PS2		1	$15	1	$15	
CTRL: MegaRAID, 4-Ch w/ 64MB Mem +10% spares	Enter. 1500-H	AMI	2	$2,200	10	$22,000	$600
					Subtotal	**$82,916**	**$21,408**
Storage Hardware							
DISK: 9GB Drive, 10K SCSI LVD, SCA + 10% spares	OSD9205-W45		1	$566	264	$149,424	Spared
CAB: Disk, 8 SCA w/ I/F cards, 0 Disks, 3U	OSM310300-L05		1	$2,118	28	$59,304	$26,880
ACC: Deskside Pedestal	OSM3000-DSK		1	$31	28	$868	
CBL: SCSI 68-pin VHD Conn's, 5 meter	CBL134-5		1	$142	30	$4,260	
CAB: Disk, 8 SCA w/ RAID Cntl'r, 0MB, 0 Disks, 3U	OSM311000-LR		1	$4,191	2	$8,382	$4,248
MEM: 32MB OSM cache	OSM1032-MEM		1	$187	2	$374	$216
PWR: 2nd Power Supply Upgrade, OSM	OSM3000-BPF		1	$392	2	$784	$264
PWR:3000 VA UPS, 3U	UPD30001-SXR		1	$1,897	1	$1,897	$1,068
PWR: Distribution Kit, 220V	SFR220-PWR		1	$42	3	$126	
CAB: Rackmount Kit for Disk Cages	OSM3000-RMK		1	$97	3	$291	
CAB: 36U x 19" x 34" Cabinet, Open	RM361934-OFT		1	$884	1	$884	
DOOR: 36U x 19", Rear	RM3619-RDR		1	$277	1	$277	
PNL: 36U x 34" Side Skins, L&R	RM3634-SDS		1	$221	1	$221	
					Subtotal	**$227,092**	**$32,676**
Server Software							
Microsoft NT Server 4.0, Enterprise Edition, incl 25 CALs		Microsoft	3	$3,999	1	$3,999	$0
Microsoft SQL Server 7.0, Enterprise Edtion, unlimited user license		Microsoft	3	$28,999	1	$28,999	$10,475
					Subtotal	**$32,998**	**$10,475**
Client Hardware							
SYS: NetServer LC3, w/1 550MHz Proc., 0MB Mem	D8592-AV		1	$1,635	3	$4,905	$5,121
MEM: 64 MB SDRAM Memory Upgrade	D6097-AV		1	$110	6	$660	
MEM: 128 MB SDRAM Memory Upgrade	D6098-AV		1	$159	6	$954	
DISK: 4GB SCSI 3.5 Internal	D4910-AV		1	$303	3	$909	$1,224
ETHERNET: 10/100TX Mbit/sec, PCI 32-bit	D5013-AV		1	$68	6	$408	
MONITOR:15-inch Color	EVG2100-P		1	$221	3	$663	
					Subtotal	**$8,499**	**$6,345**
Client Software							
Microsoft Windows 2000 Server, incl 25 CALs		Microsoft	3	$809	3	$2,427	$0
Microsoft Visual C++ Professional 6.0		Microsoft	3	$549	1	$549	$0
					Subtotal	**$2,976**	**$0**
User Connectivity							
Ethernet Switch, 8-Port 100TX TrueFast + 10% spares	NX-SW8	Netlux	4	$229	9	$2,061	spared
Ethernet Hub, 8-Port 10Base-T (8+1 ports) + 10% spares	Z85094	General	5	$27	2970	$80,190	spared
					Subtotal	**$82,251**	**$0**
					Total	**$436,732**	**$70,904**
Unisys Service Pre-Pay Discount							**($9,816)**
Western Micro Discount			1			**($29,651)**	

Notes:

1. HW Maintenance - Unisys 36 month warranty is upgraded to service level: Standard Performance-Gold. Last 24 months are also at service level: Standard Performance-Gold.
2. All Microsoft maintenance is covered by the maintenance cost of Microsoft SQL Server.
3. 10% or minimum 2 spares are added in place of onsite service (products have a five year return-to-vendor warranty)
4. Pricing: 1 = Western Micro, 2 = AMI, 3 = Microsoft, 4 = Netlux, 5 = Software House Int'l

Five Year Cost of Ownership	**$468,169**
TPC-C Throughput	**26,922.60**
$/tpmC	**$17.39**

The benchmark results and test methodology were audited by Tom Sawyer of Performance Metrics, Inc.

Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumption 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 benchmarks 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
Unisys Aquanta ES5085R Server (4P)

MQTh, Computed Maximum Qualified Throughput: **26,922.60**
% throughput difference, reported & reproducibility runs: 0.07%

Transaction Mix

New Order	44.87%
Payment	43.02%
Delivery	4.03%
Stock-Level	4.02%
Order-Status	4.03%

Response Times

Transaction	Average	Maximum	90th %ile
New-Order	0.36	4.59	0.48
Payment	0.24	4.54	0.35
Delivery	0.13	0.77	0.14
Stock-Level	1.89	3.61	2.23
Order Status	0.27	4.43	0.38
Menu	0.13	3.50	0.14
Delivery (Deferred)	0.44	1.75	0.64

Response time delay added for emulated components (seconds)

RT Response time	0.1
Menu Response time	0.1

Keying/Think Time Times (seconds)

Transaction	Minimum	Average	Maximum
New-Order	18.00/0	18/12.03	18.67/120.3
Payment	3.00/0	3/12.03	3.66/120.3
Delivery	2.00/0	2/5.04	2.52/50.6
Stock-Level	2.00/0	2/5.06	2.59/50.6
Order-Status	2.00/0	2/10.06	2.62/100.7

Test Duration

Ramp up time	58 minutes
Measurement interval (M)	20 minutes
Transactions (all types) completed during measurement interval	1,199,890
Ramp-down time	51 minutes

Checkpointing:

Number of checkpoints	1
Checkpoint interval	20 minutes

Table of Contents

Figures

Tables

Document Structure

The TPC Benchmark C Standard Specification requires test sponsors to publish, submit to the TPC, and make available to the public, a full disclosure report for any result to be considered compliant with the specification. The required contents of the full disclosure report are specified in Clause 8.

This report is submitted to satisfy the specification's requirement for full disclosure. It documents the compliance of the benchmark implementation and execution reported for the Unisys Corporation Aquanta ES5085R Server (4P) using Microsoft SQL Server 7.0, Enterprise Edition, on Microsoft Windows NT 4.0, Enterprise Edition.

TPC Benchmark C Overview

The TPC Benchmark™ C Standard Specification Revision 3.5 was developed by the Transaction Processing Performance Council (TPC). It is the intent of the TPC to develop a suite of benchmarks to measure the performance of computer systems executing a wide range of applications. Unisys and Microsoft Corporations are active participants in the TPC to define and develop such a suite of benchmarks.

TPC Benchmark™ C (TPC-C) is an OLTP workload. It is a mixture of read-only and update intensive transactions that simulate the activities found in complex OLTP application environments. It does so by exercising a breadth of system components associated with 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 on data access and update.

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

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

0. General Items

0.1. Order and Titles

The order and titles of sections in the Test Sponsor's Full Disclosure report must correspond with the order and titles of sections from the TPC-C standard specification (i.e., this document). The intent is to make it as easy as possible for readers to compare and contrast material in different Full Disclosure reports.

The order and titles of the sections in this report correspond with those from the TPC-C standard specification.

0.2. Executive Summary Statement

The TPC Executive Summary Statement must be included near the beginning of the Full Disclosure report.

The TPC Executive Summary Statement is included near the beginning of this report.

0.3. Numerical Quantities Summary

The numerical quantities listed below must be summarized near the beginning of the Full Disclosure Report :

- *measurement interval in minutes,*
- *number of checkpoints in the measurement interval,*
- *checkpoint interval in minutes,*
- *number of transactions (all types) completed within the measurement interval,*
- *computed Maximum Qualified Throughput in tpmC,*
- *percentage difference between reported throughput and throughput obtained in reproducibility run,*
- *ninetieth percentile, average and maximum response times for the New-Order, Payment, Order-Status, Stock-Level, Delivery (deferred and interactive) and Menu transactions,*
- *time in seconds added to response time to compensate for delays associated with emulated components,*
- *percentage of transaction mix for each transaction type.*

These numerical quantities are summarized near the beginning of this report.

0.4. Application Code Disclosure

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

Appendix A contains the client application code used in this TPC-C benchmark. Appendix B contains the SQL stored procedures which implement the TPC-C transactions.

0.5. Benchmark Sponsor

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

This TPC benchmark C was sponsored by Unisys Corporation. The benchmark test was developed by Microsoft and Unisys. The benchmark was conducted at Unisys, Mission Viejo, California.

0.6. Parameter Settings

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

- *Data Base tuning options*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and application configuration parameters*

Appendix C contains the configuration and system parameters used in running these TPC-C tests. It also contains all the client and server OS and SQL Server tunable parameters.

0.7. Configuration Diagrams

Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

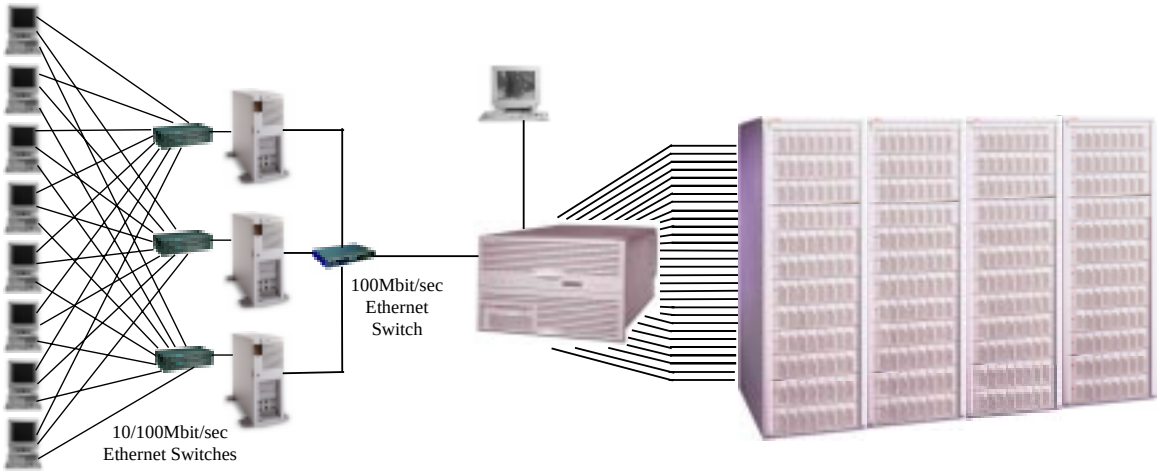
- *Number and type of processors.*
- *Size of allocated memory, and any specific mapping/partitioning of memory unique to the test.*
- *Number and type of disk units (and controllers, if applicable).*
- *Number of channels or bus connections to disk units, including their protocol type.*
- *Number of LAN (e.g., Ethernet) connections, including routers, workstations, terminals, etc., that were physically used in the test or are incorporated into the pricing structure (see Clause 8.1.8).*
- *Type and the run-time execution location of software components (e.g., DBMS, client processes, transaction monitors, software drivers, etc.).*

The Remote Terminal Emulator (RTE) software used for these TPC-C tests is proprietary to Unisys. The benchmarked configuration of the RTE and Aquanta ES5085R Server (4P) is illustrated in Figure 0.1. Tables 4.3, 4.4 and 4.5 contain a detailed explanation of the disk configuration.

The priced configuration for the Aquanta ES5085R Server (4P) is shown in Figure 0.2.

Figure 0.1: Benchmarked Configuration

Aquanta ES5085R Server (4P) - Benchmarked Configuration



8 RTEs
emulating 21,480 users
12 LANs w/ 890 users each
12 LANs w/ 900 users each

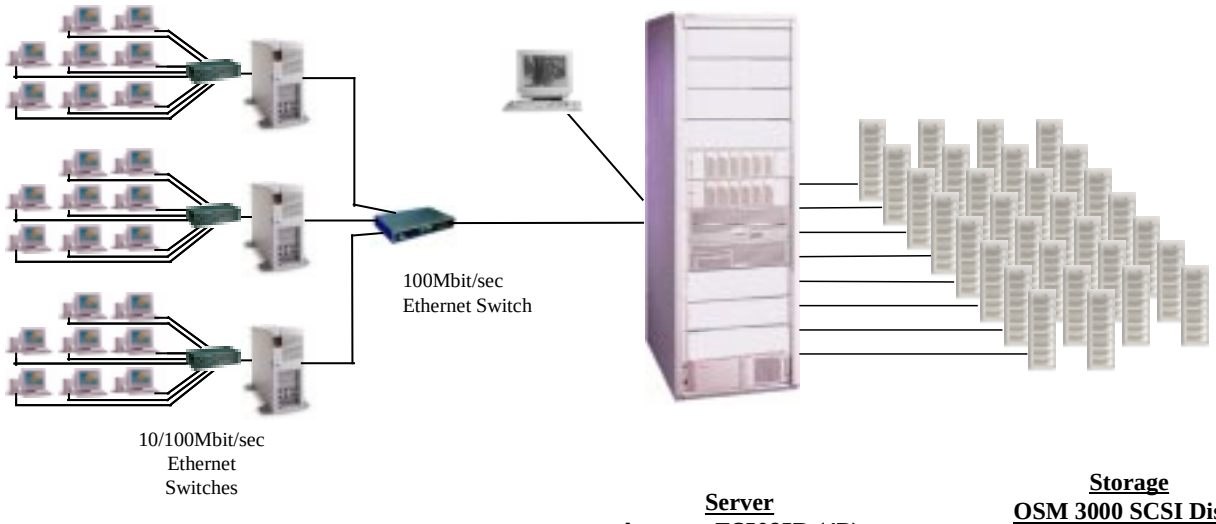
Clients (each)
NetServer LC3
1x 550MHz Pentium® III CPU,
512KB L2 cache,
384MB memory,
1 x 3.97GB disk,
2 PCI Fast Ethernet adapters for
Server & RTE connections

Server
Aquanta ES5085R (4P)
4x Pentium® III Xeon 550MHz CPUs,
2MB L2 cache per CPU,
4 GB memory,
1 Internal SCSI controller,
1 x 4.24 GB internal SCSI disk,
8 PCI SCSI RAID controllers,
1 PCI Fast Ethernet adapter

Storage
OSM 3000 SCSI Disks
224 x 8.54 GB data,
16 x 8.54 GB log,
2 x SCSI-SCSI RAID
Controllers

Figure 0.2: Priced Configuration

Aquanta ES5085R Server (4P) - Priced Configuration



21,480 PC's

Clients (each)
NetServer LC3
1x 550MHz Pentium® III CPU,
512KB L2 cache,
384MB memory,
1 x 3.97GB disk,
2 PCI Fast Ethernet adapters for
Server & PC connections

Server
Aquanta ES5085R (4P)
4x Pentium® III Xeon 550MHz CPUs,
2MB L2 cache per CPU,
4 GB memory,
1 Internal SCSI controller,
1 x 8.50 GB internal SCSI disk,
8 PCI SCSI RAID controllers,
1 PCI Fast Ethernet adapter

Storage
OSM 3000 SCSI Disks
224 x 8.54 GB data,
16 x 8.54 GB log,
2 x SCSI-SCSI RAID
Controllers,
1 Uninterruptable Power
Supply

1. Clause 1: Logical Database Design

1.1. Table Definitions

Listings must be provided for all table definition statements and all other statements used to setup the data base.

Appendix B contains the SQL definitions of all the required database files, filegroups, tables, indexes and stored procedures, plus a listing of the program used to load the database and establish the required initial populations of each table.

1.2. Physical Organization of the Database

The physical organization of tables and indices, within the data base, must be disclosed.

The disk space was allocated to SQL Server according to the data in Tables 4.3, 4.4 and 4.5. The SQL definitions are contained in Appendix B.

1.3. Insert and/or Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restriction in the SUT data base 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.

There were no restrictions on insert and/or delete operations to any of the tables.

1.4. Partitioning

While there are 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 for any table in this implementation.

1.5. Replication, Duplication or Additions

Replication of tables, if used, must be disclosed.

Additional and/or duplicate attributes in any table must be disclosed along with a statement on the impact on performance.

No replications, duplications or additional attributes were used in this implementation.

2. Clause 2: Transaction & Terminal Profiles

2.1. Random Number Generation

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

The drivers used the Unisys RTE program, which was independently audited. The initial population of the database was performed by the loader program from V4.20 of the Microsoft TPC-C toolkit, which was also independently audited. Furthermore, the auditor sampled various initial and runtime distributions produced by this implementation to verify correctness.

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 Benchmark C Standard Specification. There are some minor differences in appearance due to the use of a web client implementation.

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).

This was verified by the auditor.

2.4. Presentation Managers or Intelligent Terminal

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

Application code running on the client implemented the TPC-C user interface. A listing of this code is included in Appendix A. No presentation manager was used on the client, as screen manipulation and data input/output was handled for each user by the Microsoft Internet Explorer web browser running on each user PC.

2.5. Transaction Statistics

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

The number of items per order entered by New-Order transactions must be disclosed.

The percentage of home and remote Payment transactions must be disclosed.

The percentage of Payment and Order-Status transactions that used non-primary key (C_LAST) access to the database must be disclosed.

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

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

Table 5.4 in Section 5 contains all these statistics.

2.6. Queuing Mechanism of Delivery

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

Deferred deliveries are queued by making an entry in an array within the application process (tpcc.dll) running on the client systems. Background threads within the application process asynchronously process the queued delivery transactions and log the results to a file upon completion.

3. Clause 3: Transaction & System Properties

3.1. Transaction System Properties (ACID)

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

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).

This section defines each of these properties, describes the steps taken to ensure that they were present during the test and describes a series of tests done to demonstrate compliance with the specification. All ACID property tests were executed successfully.

3.2. Atomicity

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

3.2.1. Completed Transaction

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number) and verify that the records in the CUSTOMER, DISTRICT, and WAREHOUSE tables have been changed appropriately.

The balances from a randomly selected warehouse, district, and customer row were retrieved by customer number from a script. A Payment transaction was submitted with the same warehouse, district and customer identifiers for a known amount. After completion of the Payment transaction, the balances of the selected warehouse, district, and customer were again retrieved to verify that the changes had been made correctly.

3.2.2. Aborted Transactions

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number) and substitute a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the records in the CUSTOMER, DISTRICT, and WAREHOUSE tables have NOT been changed.

The balances from a randomly selected warehouse, district, and customer row were retrieved by customer number from a script. A Payment transaction was submitted with the same warehouse, district and customer identifiers that issued a ROLLBACK command rather than a COMMIT. After the transaction completed, the balances of the selected warehouse, district, and customer were again retrieved to verify that no changes had been made to the database.

3.3. Consistency

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

The benchmark specification requires explicit demonstration of the following four consistency conditions:

1. The sum of the district balances in a warehouse is equal to the warehouse balance;
2. For each district, the next order id minus one is equal to maximum order id in the ORDER table and equal to the maximum new order id in the NEW ORDER table;
3. For each district, the maximum order id minus minimum order id in the ORDER table plus one equals the number of rows in the NEW-ORDER table for that district;
4. For each district, the sum of the order line counts in the ORDER table equals the number of rows in the ORDER-LINE table for that district;

In order to demonstrate this consistency, the following steps were taken:

1. Prior to the start of a benchmark run, the consistency of the database was verified by testing successfully conditions 1-4 described above with a script.
2. A run under full user load was executed for over 10 minutes with a checkpoint during the run.
3. After completion of that test, the consistency of the database was again verified by successfully testing using the same consistency script as in step 1.

3.4. Isolation

Sufficient conditions must be enabled at either the system or application level to ensure the required isolation defined above (clause 3.4.1) is obtained.

The benchmark specification defines seven required tests to be performed to demonstrate that required levels of transaction isolation are met. These tests, described in Clauses 3.4.2.1 - 3.4.2.7, were all performed from a script and verified by the auditor. In Isolation Test 7, Case A was observed. In addition, the phantom tests and stock level tests were executed and verified to be successful.

3.5. Durability

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

Three durability tests were executed to satisfy the requirements of the specification. The test for loss of memory and instantaneous interruption was combined and performed with a fully scaled database with all emulated users. The loss of log and loss of data tests were performed on the same system, using a ten warehouse database with 100 emulated users. To the best of our knowledge, these tests prove that the fully scaled configuration used for the throughput test would also meet all durability tests.

3.5.1. Loss of Log Disk and Loss of Data Disk

The following steps were taken (using a ten warehouse database on the same system) to demonstrate durability in the case of loss of a log disk and of data disk. The same log disks and controllers were used for the log as for the fully scaled database. Unused space on two data controllers of the fully scaled database was used as the data area for the 10 warehouse database; unused space on a third data controller was used as backup for the small database.

1. The database was backed up to extra disks on a backup device.
2. The D_NEXT_O_ID fields for all rows in the district table were summed up to determine the initial count of orders present in the database.
3. The RTE was started with 100 users. On the driver systems, committed and rolled back New-Order transactions were recorded in a "success" file.
4. After five minutes of running at steady state, a hot-pluggable log disk was removed from the disk cabinet, with no effect on NT or SQL server.
5. After 5 additional minutes of operation, a hot-pluggable data disk was removed from the disk cabinet.
6. NT and SQL Server encountered IO errors due to the missing disk and recorded these errors in the NT event log and SQL Server error log, respectively. The RTEs also recorded errors.
7. First, the RTEs and clients were stopped, then SQL Server was used to take a dump of the transaction log to the dump device.
8. Next, scripts were executed to drop the database and all its devices. Then, SQL Server was shutdown and the SUT shutdown.
9. A data disk was inserted in the disk cabinet to replace the one removed. The RAID controller was used to recreate the stripe set containing the new data disk. (The missing log drive was not replaced.)
10. The SUT was restarted, and Disk Administrator was used to assign the proper drive letter to the new volume. SQL Server was then restarted and a new (empty) database created as part of the restore database process. That process loaded the initial database into the new database, but did not perform any recovery. Next the transaction log was restored, followed by transaction recovery. The latter step restored all committed transactions to the database.
11. Consistency condition 3 of Clause 3.3.2.3 was executed to verify database consistency.
12. Step 2 was repeated to determine the total number of orders. This number was subtracted from the count obtained previously in Step 2 to determine the number of additional orders added to the database.
13. The contents of the "success" files on the drivers were sampled to verify that the records in the "success" file for committed New-Order transactions had corresponding records in the ORDER. Moreover, the counts were matched with those obtained in step 12.

3.5.2. Instantaneous Interruption and Loss of Memory

Instantaneous interruption and loss of memory tests were combined because the loss of power erased the contents of memory. This failure was induced by removing the primary power to the System Under Test while the benchmark was executing.

1. The D_NEXT_O_ID fields for all rows in the district table were summed up to determine the initial count of orders present in the database (count1).
2. On the driver systems, committed and rolled back New-Order transaction were recorded in a "success" file.
3. The benchmark was executed at full load with all emulated users for a minimum of 10 minutes.
4. The system's primary power was then turned off.
5. After transaction failures were noted by the RTEs, the RTEs and clients were shutdown.
6. Power was restored to the SUT, the system rebooted, SQL Server was restarted, and automatic database recovery was performed. The database recovery used the transaction log to reapply all committed transactions and rollback any (in progress) uncommitted transactions, so that the database disks were correct.
7. After recovery finished, Consistency Condition of Clause 3.3.2.3 (no gaps in NO_O_ID) was executed to verify that the database was consistent..

8. Next, samples of the contents of the “success” file from the drivers were compared against corresponding rows of the ORDER table to verify that records in the “success” file for committed New-Order transactions had corresponding records in the ORDER table.
9. Finally, step 1 was repeated to determine the total number of orders (count2). Count2 minus count1 was not less than the number of committed New-Order records in the “success” file.

4. Clause 4: Scaling & Database Population

4.1. Initial Cardinality of Tables

The Cardinality (e.g., the 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 (see Clause 4.2.2 and the Auditor's attestation letter) the cardinality of the WAREHOUSE table as initially configured and the number of rows deleted must be disclosed.

The TPC-C database for this test was configured with 2160 warehouses. The cardinality of each table in the database is listed in Table 4.1

Table 4.1: Initial Cardinality of Database Table

Table	Occurrences
Warehouse	2,160
District	21,600
Customer	64,800,000
History	64,800,000
Order	64,800,000
New-Order	19,440,000
Order Line	647,996,638
Stock	216,000,000
Item	100,000

12 upper warehouses were inactive and not used while executing the measurement runs.

4.2. Constant Values

The following values were used as the constant C input values to the NURand function during Build and Run time for this implementation.

Table 4.2: Constant C for NURand

Function	Value
C_LAST (Build)	123
C_LAST (Run)	208

4.3. Database Layout

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

Tables 4.3, 4.4 and 4.5 list the distribution of the database over 224 9GB disks and the transaction log over 8 mirrored pairs of 9GB disks for the benchmark configuration. In addition, there was one disk containing Windows NT Enterprise Edition and SQL Server Enterprise Edition code and the Master database plus the paging file. The tested and priced disk configurations were identical, except that a larger 9GB boot drive was priced rather than the 4GB drive used in the measurements.

4.4. DBMS: Data Model and DBMS Interface/Access Language

A statement must be provided that describes:

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

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

Microsoft SQL Server 7.0, Enterprise Edition is a relational DBMS.

The client software interfaced to SQL Server through Stored Procedures invoked through Remote Procedure Calls embedded in the C application code. Specifically, DBLIB and TCP/IP sockets were used.

4.5. DBMS Partitions/Replications

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

No table partitioning or replication was done.

4.6. DBMS Space Requirements

Details of the 180 day space computation along with proof that the database is configured to sustain 8 hours of growth for dynamic tables (Order, Order-line, and History) must be disclosed (see Clause 4.2.3).

Appendix E lists the space requirements for the 180-day space as well as the logical log space for eight hours.

Table 4.3: Disk Cage Configuration

Disk Cage Configuration										
Adapter	**Channel**	**Id**	**Id**	**Id**	**Id**	**Id**	**Id**	**Id**	**Id**	**Rack #**
1	**0**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	1
	1	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	2
2	**0**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	3
	1	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	4
	2	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	5
	3	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	6
3	**0**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	7
	1	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	8
	2	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	9
	3	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	10
4	**0**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	11
	1	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	12
	2	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	13
	3	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	14
5	**0**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	15
	1	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	16
	2	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	17
	3	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	18

Table 4.3: Disk Cage Configuration (Continued)

Disk Cage Configuration										
Adapter	**Channel**	**Id**	**Id**	**Id**	**Id**	**Id**	**Id**	**Id**	**Id**	**Rack #**
6	**0**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	19
	1	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	20
	2	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	21
	3	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	22
7	**0**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	23
	1	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	24
	2	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	25
	3	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	26
8	**0**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	27
	1	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	28
	2	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	29
	3	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
		9GB	9GB	9GB	9GB	9GB	9GB	9GB	9GB	30

Table 4.4: RAID Adapter Disk Configuration

RAID Adapter Disk Configuration							
Adapter	ID	Channel 0	Channel 1	Channel 2	Channel 3	RAID Configuration	Drive Letters
1	0						
	1						
	2						
	3						
	4						
	5						
	6						
	8						
	9						
	10						
	11						
	12						
	13						
	14						
	15	A01-01	A01-02			Configure Array 1 as RAID 1	L:
2	8	A01-01	A02-01	A03-01	A04-01		
	9	A01-02	A02-02	A03-02	A04-02	Configure Arrays 1-4 as RAID 0	D: and E:
	10	A01-03	A02-03	A03-03	A04-03		
	11	A01-04	A02-04	A03-04	A04-04		
	12	A01-05	A02-05	A03-05	A04-05		
	13	A01-06	A02-06	A03-06	A04-06		
	14	A01-07	A02-07	A03-07	A04-07		
	15	A01-08	A02-08	A03-08	A04-08		
3	8	A01-01	A02-01	A03-01	A04-01		
	9	A01-02	A02-02	A03-02	A04-02	Configure Arrays 1-4 as RAID 0	F: and G:
	10	A01-03	A02-03	A03-03	A04-03		
	11	A01-04	A02-04	A03-04	A04-04		
	12	A01-05	A02-05	A03-05	A04-05		
	13	A01-06	A02-06	A03-06	A04-06		
	14	A01-07	A02-07	A03-07	A04-07		
	15	A01-08	A02-08	A03-08	A04-08		
4	8	A01-01	A02-01	A03-01	A04-01		
	9	A01-02	A02-02	A03-02	A04-02	Configure Arrays 1-4 as RAID 0	H: and I:
	10	A01-03	A02-03	A03-03	A04-03		
	11	A01-04	A02-04	A03-04	A04-04		
	12	A01-05	A02-05	A03-05	A04-05		
	13	A01-06	A02-06	A03-06	A04-06		
	14	A01-07	A02-07	A03-07	A04-07		
	15	A01-08	A02-08	A03-08	A04-08		
5	8	A01-01	A02-01	A03-01	A04-01		
	9	A01-02	A02-02	A03-02	A04-02	Configure Arrays 1-4 as RAID 0	J: and K:
	10	A01-03	A02-03	A03-03	A04-03		
	11	A01-04	A02-04	A03-04	A04-04		
	12	A01-05	A02-05	A03-05	A04-05		
	13	A01-06	A02-06	A03-06	A04-06		
	14	A01-07	A02-07	A03-07	A04-07		
	15	A01-08	A02-08	A03-08	A04-08		

Table 4.4: RAID Adapter Disk Configuration (Continued)

RAID Adapter Disk Configuration							
Adapter	ID	Channel 0	Channel 1	Channel 2	Channel 3	RAID Configuration	Drive Letters
6	8	A01-01	A02-01	A03-01	A04-01		
	9	A01-02	A02-02	A03-02	A04-02	Configure Arrays 1-4 as RAID 0	M: and N:
	10	A01-03	A02-03	A03-03	A04-03		
	11	A01-04	A02-04	A03-04	A04-04		
	12	A01-05	A02-05	A03-05	A04-05		
	13	A01-06	A02-06	A03-06	A04-06		
	14	A01-07	A02-07	A03-07	A04-07		
	15	A01-08	A02-08	A03-08	A04-08		
7	8	A01-01	A02-01	A03-01	A04-01		
	9	A01-02	A02-02	A03-02	A04-02	Configure Arrays 1-4 as RAID 0	O: and P:
	10	A01-03	A02-03	A03-03	A04-03		
	11	A01-04	A02-04	A03-04	A04-04		
	12	A01-05	A02-05	A03-05	A04-05		
	13	A01-06	A02-06	A03-06	A04-06		
	14	A01-07	A02-07	A03-07	A04-07		
	15	A01-08	A02-08	A03-08	A04-08		
8	8	A01-01	A02-01	A03-01	A04-01		
	9	A01-02	A02-02	A03-02	A04-02	Configure Arrays 1-4 as RAID 0	Q: and R:
	10	A01-03	A02-03	A03-03	A04-03		
	11	A01-04	A02-04	A03-04	A04-04		
	12	A01-05	A02-05	A03-05	A04-05		
	13	A01-06	A02-06	A03-06	A04-06		
	14	A01-07	A02-07	A03-07	A04-07		
	15	A01-08	A02-08	A03-08	A04-08		

Table 4.5: Disk Administrator Configuration

Disk Administrator Configuration			
Disk 0 69982 MB	L: unknown 68805 MB	 unknown 1177 MB	 free space 0 MB
Disk 1 279904 MB	D: unknown 18005 MB	E: unknown 9505 MB	 free space 252394 MB
Disk 2 279904 MB	F: unknown 18005 MB	G: unknown 9505 MB	 free space 252394 MB
Disk 3 279904 MB	H: unknown 18005 MB	I: unknown 9505 MB	 free space 252394 MB
Disk 4 279904 MB	J: unknown 18005 MB	K: unknown 9505 MB	 free space 252394 MB
Disk 5 279904 MB	M: unknown 18005 MB	N: unknown 9505 MB	 free space 252394 MB
Disk 6 279904 MB	O: unknown 18005 MB	P: unknown 9505 MB	 free space 252394 MB
Disk 7 279904 MB	Q: unknown 18005 MB	R: unknown 9505 MB	 free space 252394 MB
Disk 8 4338 MB	C: SYSTEM FAT 2047 MB	Z: testfiles NTFS 2291 MB	 free space 0 MB
CD-ROM 0	D:		

5. Clause 5: Performance Metrics & Response Time

5.1. Measured Throughput (tpmC)

Measured tpmC must be reported.

The measured tpmC was 26,922.60.

5.2. Response Times

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

Table 5.1: Response Time Data

Transaction	Average	Maximum	90th %ile
New-Order	0.36	4.59	0.48
Payment	0.24	4.54	0.35
Delivery	0.13	0.77	0.14
Stock-Level	1.89	3.61	2.23
Order Status	0.27	4.43	0.38
Menu	0.13	3.50	0.14
Delivery (Deferred)	0.44	1.75	0.64

5.3. Keying and Think Times

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

Table 5.2: Keying Times

Transaction	Minimum	Average	Maximum
New-Order	18.00	18.00	18.67
Payment	3.00	3.00	3.66
Delivery	2.00	2.00	2.52
Stock-Level	2.00	2.00	2.59
Order Status	2.00	2.00	2.62

Table 5.3: Think Times

Transaction	Minimum	Average	Maximum
New-Order	0.00	12.03	120.30
Payment	0.00	12.03	120.30
Delivery	0.00	5.04	50.60
Stock-Level	0.00	5.06	50.60
Order Status	0.00	10.06	100.70

5.4. Response Time Frequency Distribution Curves

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

Figure 5.1: New Order Response Time Distribution

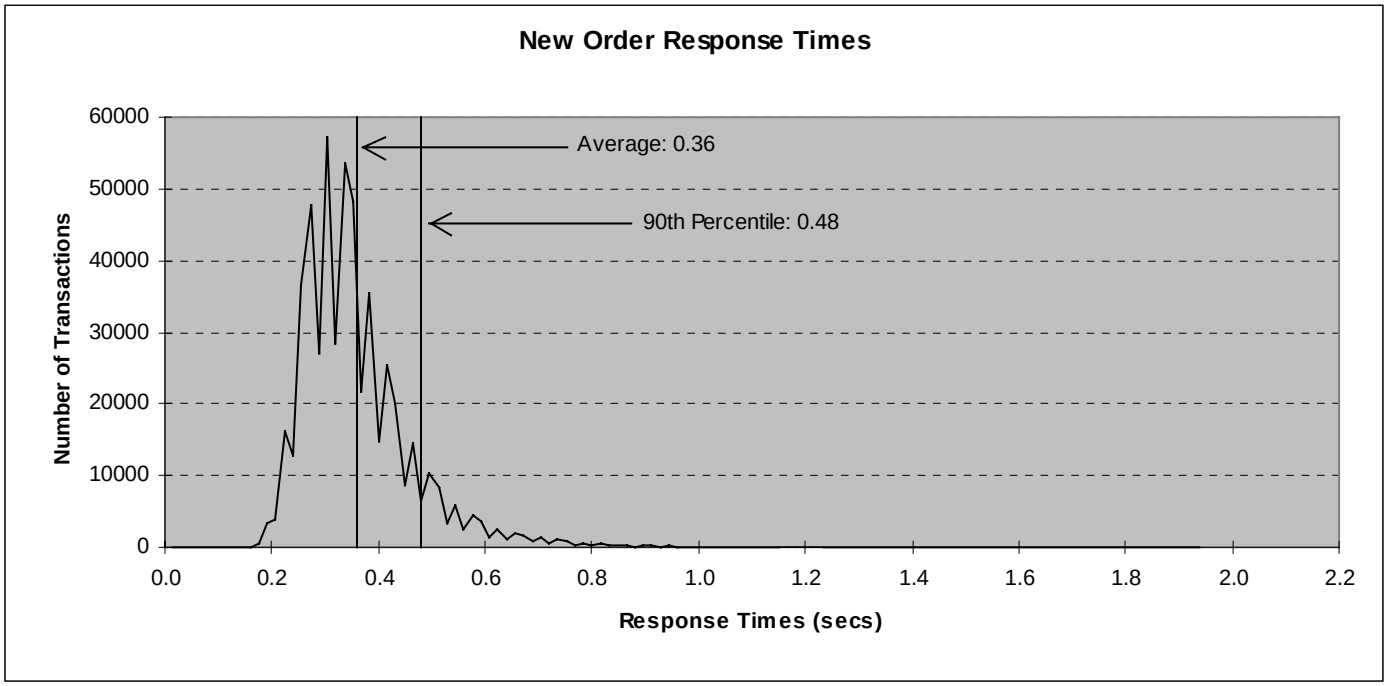


Figure 5.2: Payment Response Time Distribution

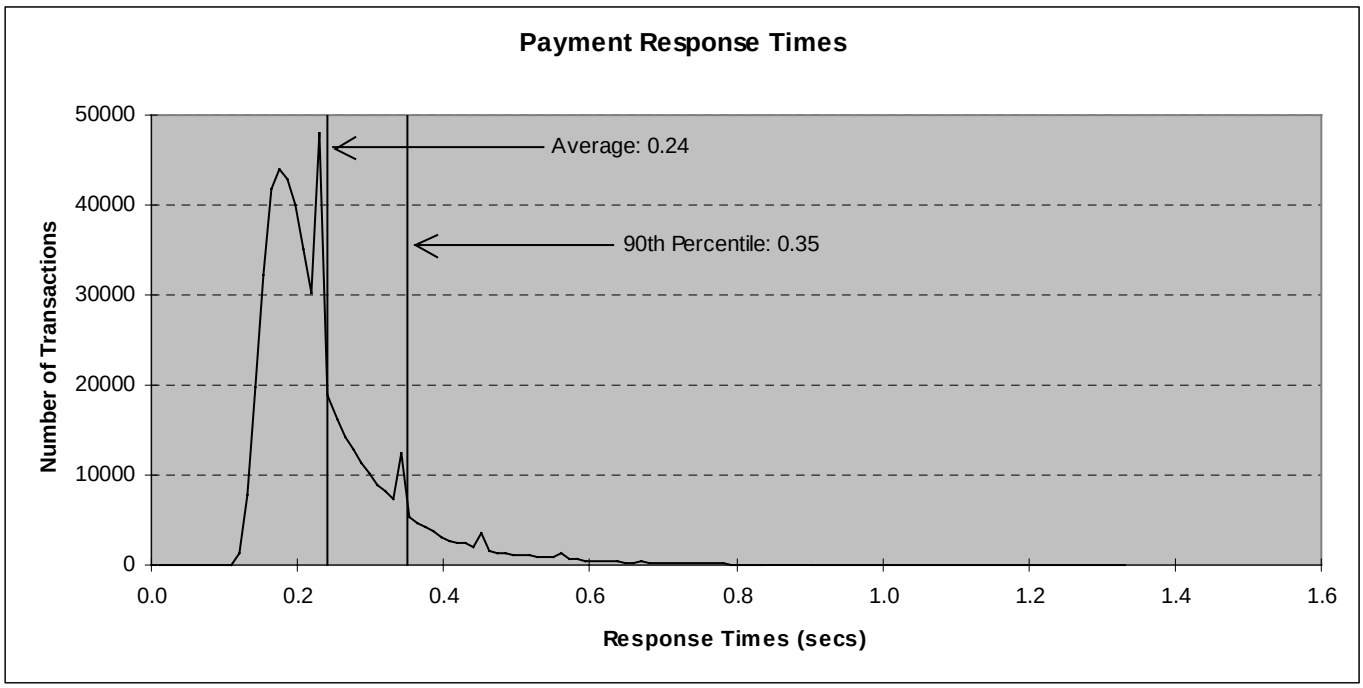


Figure 5.3: Order Status Response Time Distribution

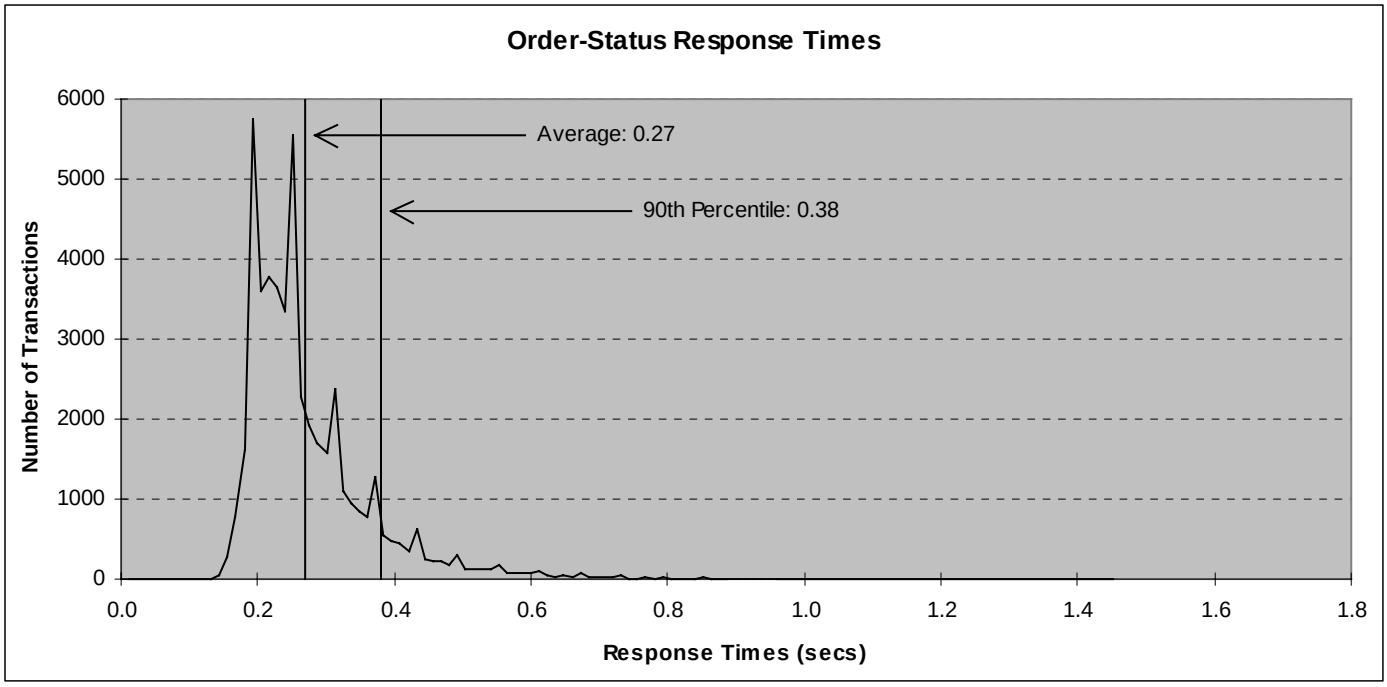


Figure 5.4: Delivery Response Time Distribution

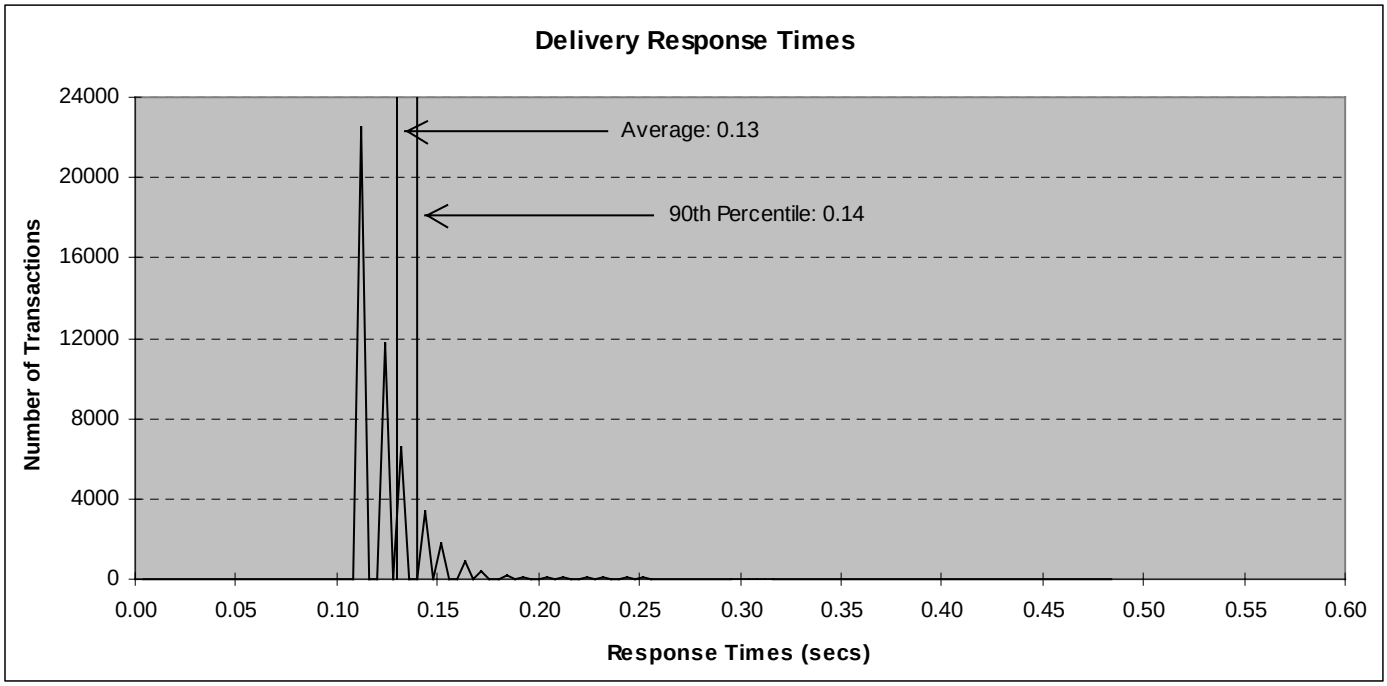
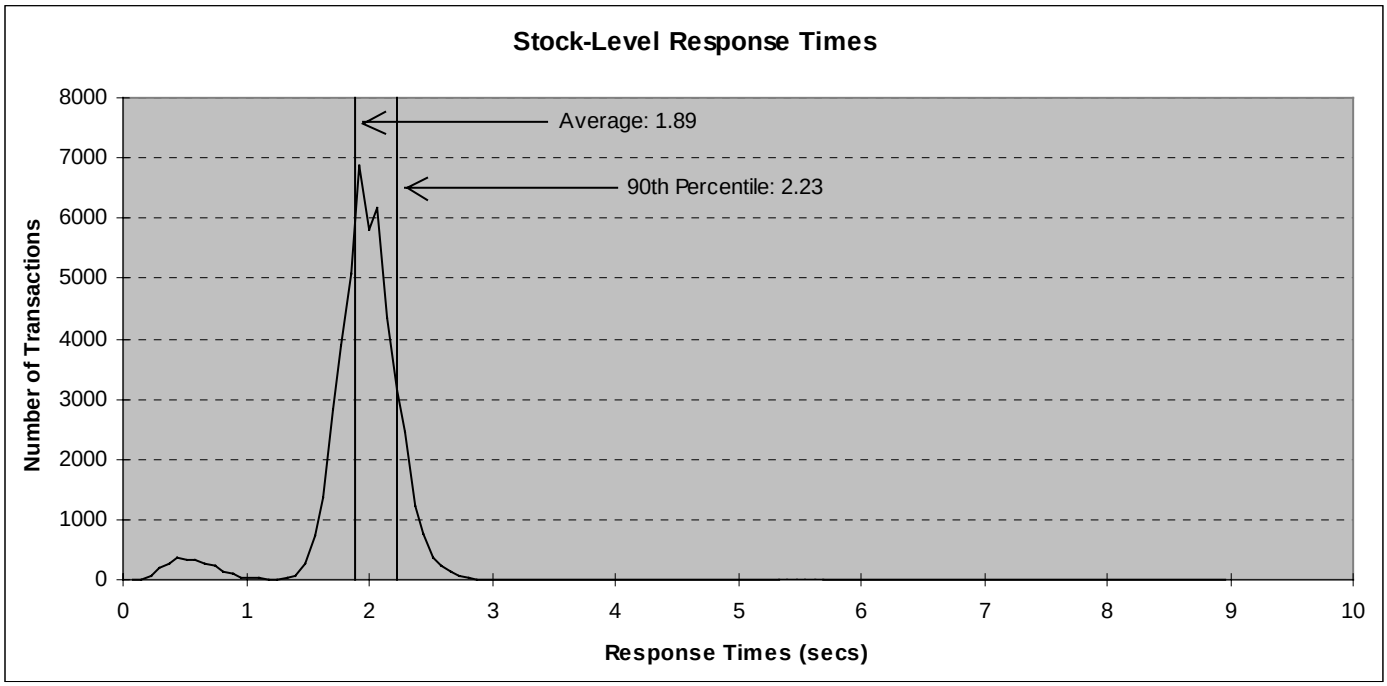


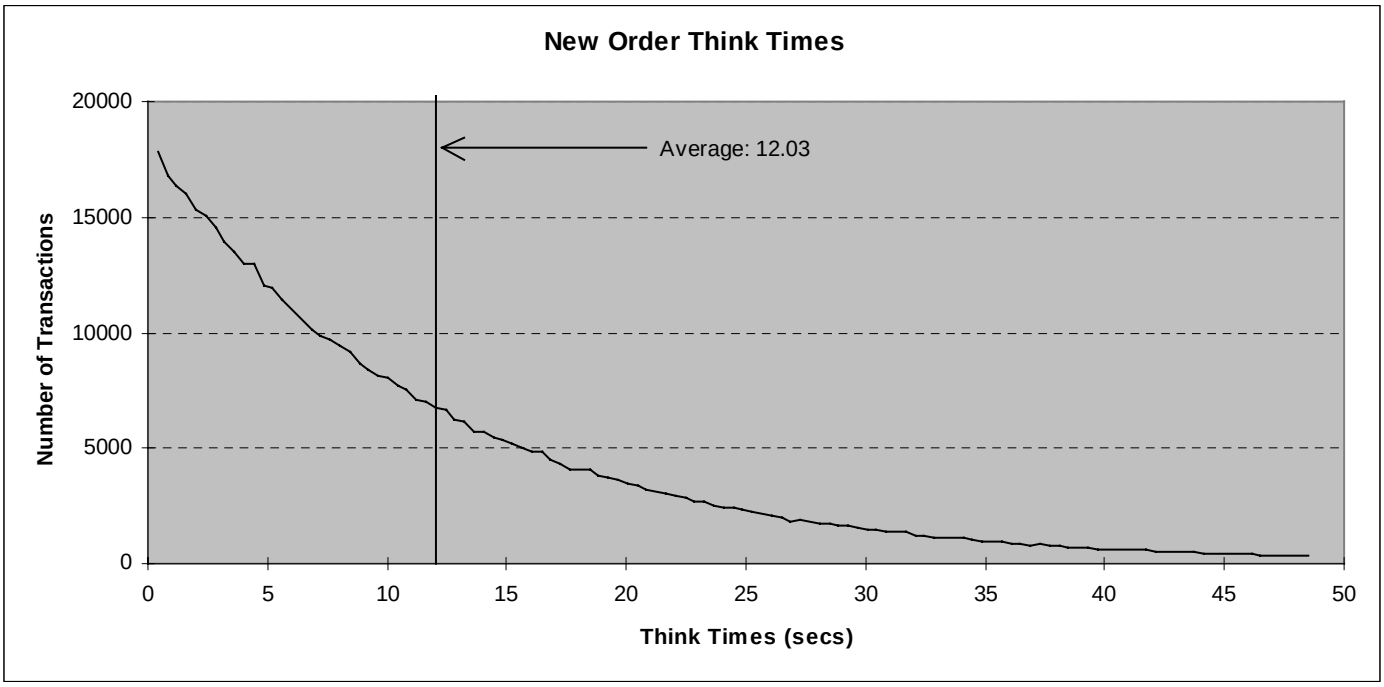
Figure 5.5: Stock Level Response Time Distribution



5.5. New Order Think Time Frequency Distribution Curve

Think Time frequency distribution curve (see Clause 5.6.3) must be reported for the New-Order transaction.

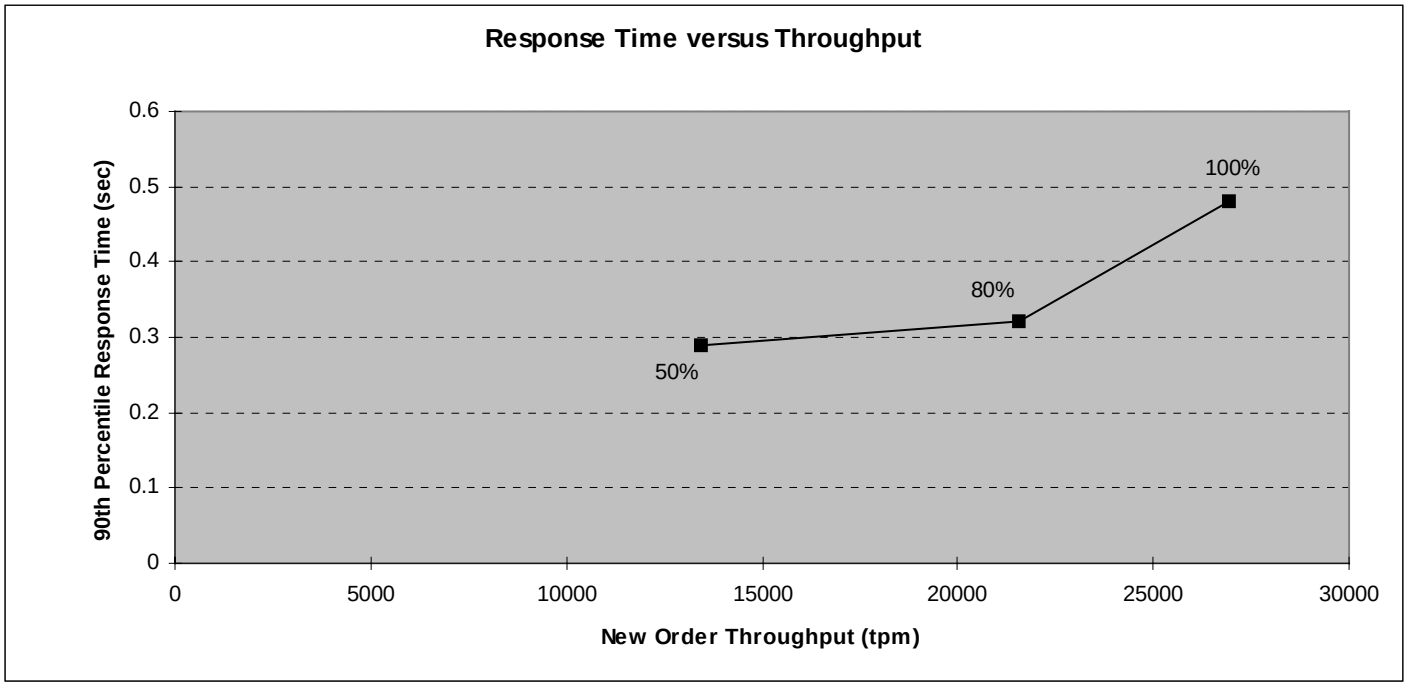
Figure 5.6: New Order Think Time Distribution



5.6. Response Time versus Throughput Performance Curve

The performance curve for response times versus throughput (Clause 5.6.2) must be reported for the New-Order transaction

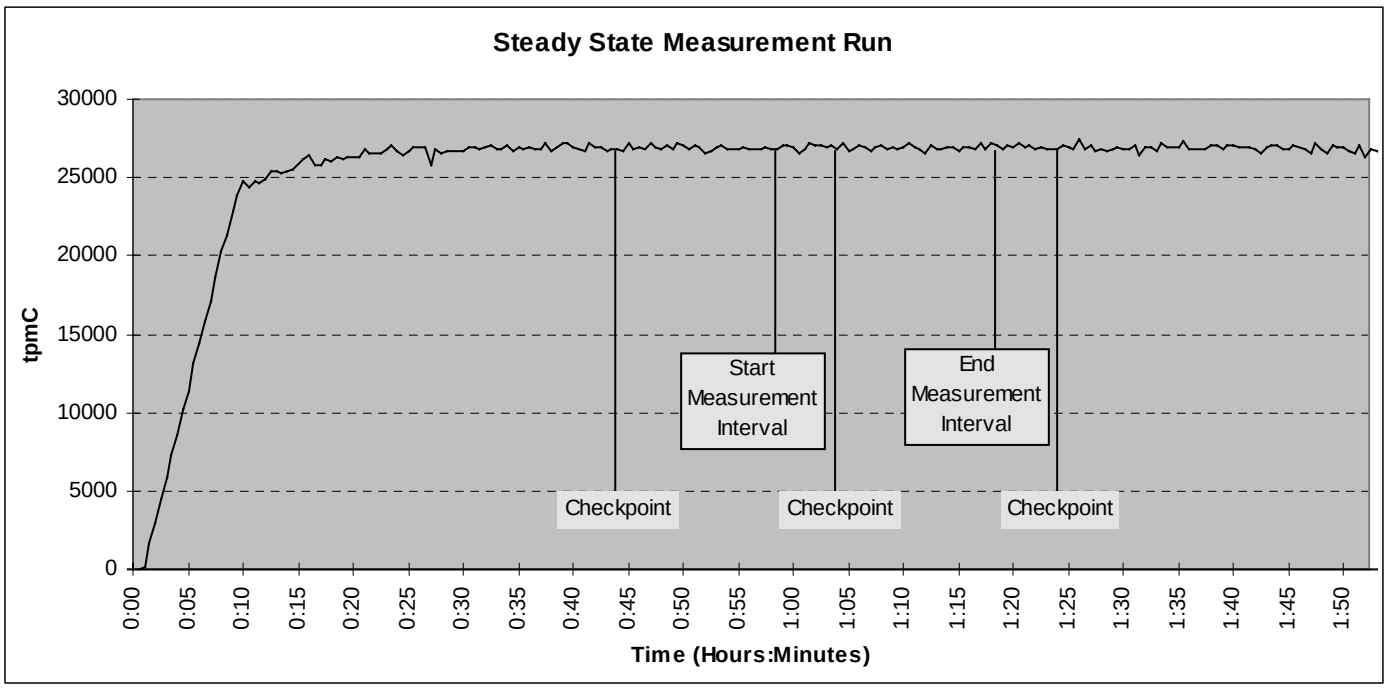
Figure 5.7: Response Time versus Throughput



5.7. New-Order Throughput vs. Time

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

Figure 5.8: Throughput (tpmC) versus Time



5.8. Determination of "Steady State"

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

The transaction throughput rate (tpmC) and response time were relatively constant after the initial 'ramp up' period. The throughput and response time behavior were determined by examining data reported for each 30-second interval over the duration of the benchmark. Ramp-up and steady state are discernible in the graph presented in Figure 5.8.

5.9. 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 RTE selects a transaction type from the menu and prepares to request the appropriate blank form. A timestamp is taken before the form request is sent and after the response is returned. The difference between the two is saved off as the menu response time. The RTE then generates input data for the transaction to create a completed form and waits the appropriate key time. A timestamp is taken before the completed form is sent and after the response is returned. The difference between these two is saved off as the transaction response time. Both response times are padded with a 0.1 second delay per spec to account for the web browser delay. The appropriate transaction data and response times are logged and the RTE waits the required think time interval before repeating the process. Each RTE driver maintains its own log file. Log file contents are consolidated for the reports.

The RTE emulates web browsers (not terminals) in this client-server implementation. The RTE sends and receives HTML formatted data using HTTP through Ethernet LANs to a client application running on the client machine. The client application processes the request, sends the transaction to a COM+ component, waits for the transaction response, and returns an appropriately formatted HTML form back to the (emulated) web browser (RTE). When activated, the COM+ component calls a stored procedure on the data base server using Microsoft SQL Server DBLIB and RPC through sockets, collects the response, and returns the result to the requestor.

To perform checkpoints at specific intervals, SQL Server's checkpoint interval was set to the maximum allowable value and a utility was written to schedule checkpoints at parameter-specified intervals and record the start and end time of each checkpoint. The checkpoint script was started manually on one of the client machines after the RTE had all users logged in and sending transactions and a steady state had been achieved. Using this information, the positioning of the checkpoint within the measurement interval was verified to be clear of the guard zones.

At each checkpoint, SQL Server wrote to disk all database pages in memory that had been updated but not yet physically written to the disk. Upon completion of the checkpoint, SQL Server also wrote records to the error log indicating that a checkpoint had completed.

5.10. Reproducibility

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

In a repeat test, carried out in the same manner as the primary test, a throughput of 26,905.10 tpmC was achieved on the same database during a 20-minute, steady state run. All required transaction statistics were met. See the Auditor's attestation letter for details.

5.11. Measurement Interval Duration

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

The measurement interval was 20 minutes.

5.12. 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 RTE was given a weighed random distribution which could not be adjusted during the run.

5.13. Transaction Statistics

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

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 remote Payment transactions must be disclosed.

The percentage of customer selections 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 5.4 shows this information.

Table 5.4: Transaction Statistics

Transaction Type	Statistics	Value
New Order	Rolledback transactions	0.99%
	Home warehouse	99.00%
	Remote warehouse	1.00%
	Average Items per Order	10.00
Payment	Home warehouse	84.92%
	Remote warehouse	15.08%
	Non-primary key access	59.99%
Order Status	Non-primary key access	60.25%
Delivery	Skipped transactions (Interactive)	0
	Skipped transaction counts (Deferred)	0
	Skipped District counts (Deferred)	0
Transaction Mix	New Order	44.87%
	Payment	43.02%
	Delivery	4.03%
	Stock-Level	4.02%
	Order-Status	4.03%

5.14. Checkpoint Statistics

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.

There is one checkpoint in the measurement interval. The checkpoint starts 332 seconds into the measurement interval. The checkpoint interval is 20 minutes (from the start of one to the start of the next) and a checkpoint lasts approximately 3.8 minutes. In conformance with Clause 5.5.2.2, the checkpoint occurs outside the guard zones.

6. Clause 6: SUT, Driver & Communications Definition

6.1. Remote Terminal Emulator (RTE) Description

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

The RTE used is proprietary to Unisys. Appendix D contains the profile used as input to this RTE.

6.2. Emulated Components

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

There were no emulated components in the benchmark configuration other than the emulated web browsers on the users' PCs.

6.3. Functional Diagrams

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

Section 0.7 describes and shows functional diagrams of the benchmarked and priced systems.

6.4. Network Configuration

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

Figures 0.1 and 0.2 in Section 0.7 also diagram the network configurations of the benchmark and configured systems and represent the RTEs connected via LAN replacing the user PCs that are directly connected via LAN.

6.5. Network Bandwidth

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

Ethernet local area networks (LAN) are used in the priced and tested configurations. The database server (SUT) contains a single 10/100 megabit per second LAN adapter. This LAN segment runs at 100 megabits per second in both the priced and tested configurations. The clients contain two 10/100 megabit per second LAN adapters. Both LAN segments run at 100 megabits per second in the priced and tested configurations. 24 (8 per client) user LAN segments run at 10 megabits per second in both the priced and tested configurations. Two concatenated eight-port 10/100 megabit per second switches per client were used to connect the clients to the users. An additional eight-port 10/100 megabit per second switch was used to connect the clients to the database server. In the priced configuration,

the clients are connected to workstations (PCs running web browsers). In the tested configuration, the clients are connected to RTE driver systems emulating web browsers.

6.6. Operator Intervention

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

No operator intervention was required to sustain eight hours of operation at the reported throughput.

7. Clause 7: Pricing

7.1. 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 date. If package-pricing is used, vendor part number of the package and a description uniquely identifying each of the components of the package must be disclosed. Pricing source(s) and effective date(s) must also be reported.

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

System pricing should include subtotals for the following components: Server Hardware, Server Software, Client Hardware, Client Software, and Network Components used for terminal connection (see Clause 7.2.2.3). Clause 6.1 describes the Server and Client components.

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

A detailed list of hardware and software components along with their part numbers and prices are given in the Executive Summary near the beginning of this document.

7.1.1. System Pricing

Each priced configuration consists of an integrated system package, additional options, and components. Prices for all products are US list prices. A three year warranty is standard with this class of Unisys server products.

7.1.2. Maintenance Pricing

The five year support pricing for Unisys Corporation Open Business Server products is based on a 36-month warranty on hardware, upgraded to service level Performance-Gold, plus an additional 24 months of support at service level Performance-Gold. Microsoft support pricing is based on 5 years of annual support costs.

Unisys Corporation Standard Performance-Gold Support: four hour maximum response, onsite support for hardware provides service from 8:00 A.M. to 5:00 P.M., Monday through Friday. Service requests made as late as 5:00 P.M. will receive a response the same day.

Server disks are covered by Western Micro's 5 year, seven day return-to-factory warranty, and appropriate spares are included in the priced configuration. American Megatrend's 3 year, seven day return-to-factory warranty is extended to 5 years, and appropriate spares and upgrade price are included in the priced configuration. Netlux and Sofware House International provide 5 year, seven day return-to-factory warranties, and appropriate spares are included in the priced configuration.

7.1.3. Discounts

Unisys provides a standard pre-pay discount for maintenance service of the client, server and storage components of the priced configuration.

Western Micro provides a standard dollar-volume discount to the client, server and storage components of the priced configuration.

7.2. Availability

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.

The hardware, software and support/maintenance products priced in this benchmark are detailed on page vi.

All hardware components are available now. Microsoft Windows 2000 Server and SQL Server 7.0, Enterprise Edition SP2 will be available by December 31, 1999 (see Microsoft price quote in Appendix F).

7.3. Measured tpmC, Pricing, Price/Performance, and Availability Date

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

The measured tpmC, plus pricing calculations, price/performance, and availability are shown on pages v and vi.

7.4. Country-Specific Pricing

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

None.

7.5. 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:

- One (1) Microsoft Windows NT Server 4.0, Enterprise Edition license
- One (1) Microsoft SQL Server 7.0, Enterprise Edition license
- One (1) Microsoft Windows 2000 Server License
- One (1) Microsoft Visual C++ Professional 6.0

Microsoft SQL Server & Internet Information Server were priced for an unlimited number of users.

8. Clause 8 : Full Disclosure Availability

8.1. Availability

The Full Disclosure Report must be readily available to the public at a reasonable charge, similar to charges for similar documents by that test sponsor.

Copies of this Full Disclosure Report may be downloaded from the Transaction Processing Performance Council web site at www.tpc.org or obtained by contacting:

TPC Benchmark Administrator
Systems Analysis, Modeling & Measurement Group
Unisys Corporation, M/S 262
25725 Jeronimo Road
Mission Viejo, CA 92691
USA

9. *Clause 9 : Audit*

9.1. Auditor's Report

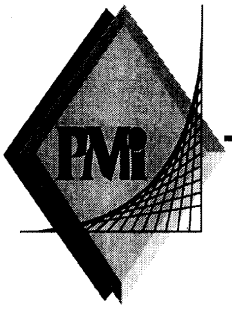
The auditor's name, address, phone number and a copy of the auditor's attestation letter indicating compliance must be included in the Full Disclosure Report.

This implementation of the TPC Benchmark C on the Unisys Aquanta ES5085R Server (4P) was audited by Tom Sawyer, a TPC certified auditor of:

Performance Metrics, Inc.
137 Yankton St. Suite 101
Folsom, CA 95630

Phone: (916) 985-1131
Fax: (916) 985-1185
e-mail: Lorna@PerfMetrics.com

The attestation letter is shown on the next 2 pages.



PERFORMANCE METRICS INC.
TPC Certified Auditors

October 26, 1999

Jerrold Buggert
Director of Modeling and Measurement
Unisys Corporation
25725 Jeronimo Road
Mission Viejo, CA 92691

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

Platform:	Unisys Aquanta ES5085R Server (4P)
Database Manager:	Microsoft SQL Server Enterprise Edition 7.0
Operating System:	Microsoft Windows NT Server Enterprise Edition 4.0 (SP4)
Transaction Manager:	Microsoft COM+ (included in Windows 2000)

Server: Aquanta ES5085R Server (4P)				
CPU's	Memory	Disks	90% Response	tpmC
4 Pentium III Xeon @ 550 Mhz	Main: 4 GB Cache: 2MB each	1 @ 4.3 GB 240 @ 9.1 GB	**0.48 sec**	**26,922.60**
3 Clients: NetServer LC3 (each)				
1 Pentium III @ 550 MHz	Main: 384 MB	1 @ 3.97 GB	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 was properly sized and populated.
- The database was properly scaled with 2,160 warehouses. There were 2,148 warehouses used in the measurement. I verified that d_next_o_id and w_ytd did not change for the unused warehouses.

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

- The ACID properties were met.
- The durability data loss and log loss tests were performed on a 10-warehouse database.
- Input data was generated according to the specified percentages.
- Eight hours of mirrored log space was configured on the measured system.
- Eight hours of dynamic table growth space was configured on the measured system.
- The 180-day space calculation was verified. The measured configuration has sufficient storage to satisfy this requirement.
- Measurement cycle times included a 0.1 second menu and a 0.1 second response time delay for an emulated Web browser.
- There were 21,480 user contexts present on the system.
- Each emulated user started with a different random number seed.
- The NURand constants used for database load and at run time were 123 and 208.
- The steady state portion of the test was 20 minutes.
- One checkpoint was taken before the measured interval.
- One checkpoint was taken during the measured interval.
- The checkpoints were verified to be clear of the guard zone.
- The system pricing was checked for major components and maintenance.

Auditor Notes:

The priced configuration substituted a 9GB drive for the 4.3GB drive. This drive contained the operating system and DBMS software. It had little to no activity during the measurement.

Sincerely,

Tom Sawyer

Tom Sawyer
Auditor

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

Appendix A - Client Source

tpcc.def

```
EXPORTS
   GetExtensionVersion
   HttpExtensionProc
```

tpcc.h

```
// tpcc.h
//
// Copyright Unisys, 1999

#include <time.h>

#define VERSIONINFO "--- 1.1.a ---"

// TPCCHandler return codes
#define TPCCSEND 1
#define TPCCSENDEND 2
#define TPCCENDNOW 3

// TPCC Service return codes
#define SVC_BADITEMID 1
#define SVC_NOERROR 0
#define SVCERR_DEADLOCK -1
#define SVCERR_NOCUSTOMER -2
#define SVCERR_NOORDERS -3
#define SVCERR_DBLIB -4
#define SVCERR_EXCEPTION -5
#define SVCERR_DQFULL -6
#define SVCERR_DQSTART -7

// Min/Max transaction data definitions
#define MIN_DId 1
#define MAX_DId 10
#define MIN_OL 5
#define MAX_OL 15
#define MIN_QUANTITY 1
#define MAX_QUANTITY 10
#define MIN_ITEM_ID 1
#define MAX_ITEM_ID 100000
#define MIN_CUST_ID 1
#define MAX_CUST_ID 3000
#define MIN_CARRIER 1
#define MAX_CARRIER 10
#define MIN_THRESHOLD 10
#define MAX_THRESHOLD 20

// pTPCC->iStatusId codes
#define INVALID_IID             1
#define STATUS_OK               0
#define ERR_CMD_UNKNOWN         -10
#define ERRTXT_CMD_UNKNOWN      "Unrecognized Command"
#define ERR_ALREADY_LOGGEDIN    -11
#define ERRTXT_ALREADY_LOGGEDIN "Already Logged In"
#define ERR_TERMID              -12
#define ERRTXT_TERMID           "TermId or SyncId in Error"
#define ERR_FORM_UNKNOWN        -13
#define ERRTXT_FORM_UNKNOWN     "Unrecognized FormId"
#define ERR_WID_INVALID         -14
#define ERR_DID_INVALID         -15
#define ERR_MISSING_KEY         -16
#define ERR_NOT_NUMERIC         -17
#define ERR_THRESHOLD_RANGE     -18
#define ERR_EMBEDDED_EMPTY_OL   -19
#define ERR_QUANTITY_INVALID    -20
#define ERR_OL_INVALID          -21
#define ERR_OL_COUNT            -22
#define ERR_TM_INTERFACE        -23
#define ERR_SERVICE_RSLT        -24
#define ERR_INPUT_TOOLONG       -25
#define ERR_IDANDNAME_EMPTY     -26
#define ERR_IDANDNAME_ENTERED   -27
#define ERR_AMOUNT_BADFORM      -28
#define ERR_AMOUNT_INVALID      -29
#define ERR_CARRIER_INVALID     -30
#define ERR_TERM_ALLOC          -31

#define STATUS_LEN 30
#define NAME_LEN 16
#define ADDR_LEN 20
#define STATE_LEN 2
#define ZIP_LEN 9

#define MAX_MSG_SZ  5000
#define CTEXT "Content-length: "
#define HTTPHdr "Content-type: text/html\r\nContent-length:     \r\n\r\n"

typedef struct
{
   unsigned short year;
   unsigned short month;
   unsigned short day;
   unsigned short hour;
   unsigned short minute;
   unsigned short second;
} DATEDETAILS;

typedef struct
{
   short ol_supply_w_id;
   long ol_i_id;
   char ol_i_name[25];
   short ol_quantity;
   char ol_brand_generic[2];
   double ol_i_price;
   double ol_amount;
   short ol_stock;
```

```
} OL_NEW_ORDER_DATA;

typedef struct
{
   bool bTPRslt;
   short iTPRslt;
   short w_id;
   short d_id;
   long        c_id;
   short o_ol_cnt;
   char c_last[NAME_LEN + 1];
   char c_credit[3];
   double c_discount;
   double w_tax;
   double d_tax;
   long o_id;
   short o_all_local;
   short o_commit_flag;
   DATEDETAILS o_entry_d;
   double total_amount;
   char execution_status[STATUS_LEN];
   OL_NEW_ORDER_DATA Ol[MAX_OL];
} NEW_ORDER_DATA;

typedef struct
{
   bool bTPRslt;
   short iTPRslt;
   short w_id;
   short d_id;
   long c_id;
   short c_d_id;
   short c_w_id;
   double h_amount;
   DATEDETAILS h_date;
   char w_street_1[ADDR_LEN + 1];
   char w_street_2[ADDR_LEN + 1];
   char w_city[ADDR_LEN + 1];
   char w_state[STATE_LEN + 1];
   char w_zip[ZIP_LEN + 1];
   char d_street_1[ADDR_LEN + 1];
   char d_street_2[ADDR_LEN + 1];
   char d_city[ADDR_LEN + 1];
   char d_state[STATE_LEN + 1];
   char d_zip[ZIP_LEN + 1];
   char c_first[NAME_LEN + 1];
   char c_middle[3];
   char c_last[NAME_LEN + 1];
   char c_street_1[ADDR_LEN + 1];
   char c_street_2[ADDR_LEN + 1];
   char c_city[ADDR_LEN + 1];
   char c_state[STATE_LEN + 1];
   char c_zip[ZIP_LEN + 1];
   char c_phone[16];
   DATEDETAILS c_since;
   char c_credit[3];
   double c_credit_lim;
   double c_discount;
   double c_balance;
   char c_data[200+1];
   char execution_status[STATUS_LEN];
} PAYMENT_DATA;

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

typedef struct
{
   bool bTPRslt;
   short iTPRslt;
   short w_id;
   short d_id;
   long c_id;
   char c_first[NAME_LEN + 1];
   short o_ol_cnt;
   char c_middle[3];
   char c_last[NAME_LEN + 1];
   double c_balance;
   long o_id;
   DATEDETAILS o_entry_d;
   short o_carrier_id;
   OL_ORDER_STATUS_DATA OlOrderStatusData[MAX_OL];
   char execution_status[STATUS_LEN];
} ORDER_STATUS_DATA;

typedef struct
{
   bool bTPRslt;
   short iTPRslt;
   short w_id;
   short o_carrier_id;
   long o_id[10];
   SYSTEMTIME QTime;                // time delivery was queued
   SYSTEMTIME EndTime;              // time delivery completed
} DELIVERY_DATA;

typedef struct
{
   bool bTPRslt;
   short iTPRslt;
   short w_id;
   short d_id;
   short thresh_hold;
   long low_stock;
   char execution_status[STATUS_LEN];
} STOCK_LEVEL_DATA;
```

tpcc.cpp

```
// tpcc.cpp
//
// Copyright Unisys, 1999
//
```

```
#include <windows.h>
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include <winreg.h>
#include <httpext.h>

#include "..\tpccsvr\tpcc.h"
#include "tmon.h"
#include "diagio.h"
#include "term.h"
#include "delivery.h"
#include "tpcchandler.h"

#define EXTN_VERSION MAKELONG(HSE_VERSION_MINOR,HSE_VERSION_MAJOR)
#define TLS_NULL 0xFFFFFFFF
DWORD dwTlsInx;
CHAR * pTitle = "IIS TPCC COM DLL";
CRITICAL_SECTION csDllMain;

// Diagnostic logging settings
BOOL bSetEventLog = TRUE;
BOOL bSetConsole = FALSE;
UINT uSetDiagLevel = DIAG_INFO;

// TMon Interface Settings
INT iTMMaxMsg = 0;

// Term Interface Settings
INT iMaxTerms = 3000;

// Delivery Settings
long lSetDThreads = 8;
long lSetDQSize = DEFAULTDQSIZE;
char szSetPath[200] = "\\inetpub\\wwwroot\\";

static CHAR * szTPCCError =
   HTTPHdr "<HTML>"
   "<HEAD><TITLE>Welcome To TPC-C</TITLE></HEAD><BODY>"
   "<B>TPCC Extension Error (TPCC Array Not Allocated)</B><BR>"
   "</BODY></HTML>";

static CHAR * szTMInitError =
   HTTPHdr "<HTML>"
   "<HEAD><TITLE>Welcome To TPC-C</TITLE></HEAD><BODY>"
   "<B>TPCC Extension Error (TMInit Failed)</B><BR>"
   "</BODY></HTML>";
INT iHHdrLen = 0;
INT iCTextLen = 0;

BOOL ThreadAttach(TPCC_STATE * pTPCC,CHAR * pDiag);
VOID ThreadDetach(TPCC_STATE * pTPCC);
VOID SendResponse(EXTENSION_CONTROL_BLOCK * pECB,CHAR * pMsg,CHAR *
pWork);
BOOL ReadRegistry(VOID);

//====================================================================
//
// Function name: DllMain
//
//====================================================================
BOOL APIENTRY DllMain(HANDLE hInst, ULONG ul_reason_for_call,
                      LPVOID lpReserved)
{
   TPCC_STATE * pTPCC = NULL;
   CHAR szDiag[MAX_DIAG_SZ];
   UINT iTMMaxSz =0;
   switch(ul_reason_for_call)
   {
      case DLL_PROCESS_ATTACH:
         // Process initialization

         InitializeCriticalSection(&csDllMain);
         ReadRegistry();
         DiagIoInit(pTitle,bSetConsole,bSetEventLog,uSetDiagLevel);
         wsprintf(szDiag,
            "(%s) EventLog = %d, Console = %d, DiagLevel = %d\n"
            "MaxTerms = %d\n",
            VERSIONINFO,bSetEventLog,bSetConsole,uSetDiagLevel,iMaxTerms);
         DiagIoWrite(szDiag,DIAG_FORCE);
         dwTlsInx = TlsAlloc();
         if (dwTlsInx == TLS_NULL)
         {
            wsprintf(szDiag,"PAttach(%ld): Tls Alloc Failed (%ld)\n",
               GetCurrentThreadId(),GetLastError);
            DiagIoWrite(szDiag,DIAG_ERROR);
            return(FALSE);
         };
         if (TermInit(iMaxTerms))
            return(FALSE);
         iTMMaxSz = max(iTMMaxSz,sizeof(NEW_ORDER_DATA));
         iTMMaxSz = max(iTMMaxSz,sizeof(PAYMENT_DATA));
         iTMMaxSz = max(iTMMaxSz,sizeof(ORDER_STATUS_DATA));
         iTMMaxSz = max(iTMMaxSz,sizeof(DELIVERY_DATA));
         iTMMaxSz = max(iTMMaxSz,sizeof(STOCK_LEVEL_DATA));
         iTMMaxSz += 10;
         TMonInit(iTMMaxSz);
         if (DeliveryInit(lSetDThreads,lSetDQSize,szSetPath))
         {
            DeliveryTerm();
            return(FALSE);
         };
         iHHdrLen = strlen(HTTPHdr);
         iCTextLen = strlen(CTEXT);
         break;
      case DLL_THREAD_ATTACH:
         // Move ThreadAttach call to HttpExt since the DllMain call
         // for Thread Attach did not reliably come before the first
         // call to HttpExtProc.
         break;
               case DLL_THREAD_DETACH:
         ThreadDetach(pTPCC);
                        break;
               case DLL_PROCESS_DETACH:
         ThreadDetach(pTPCC);
         DeleteCriticalSection(&csDllMain);
         DeliveryTerm();
         TMonTerm();
         TermTerm();
         TlsFree(dwTlsInx);
         dwTlsInx = TLS_NULL;
```

```
         DiagIoTerm();
                     break;
       };
       return TRUE;
};   // DllMain

//============================================================================
//
// Function name: ThreadAttach
//
// Result:
//    FALSE Thread state structure initialized
//    TRUE  Thread state structure initialization failure
//
//============================================================================
BOOL ThreadAttach(TPCC_STATE * pTPCC,CHAR * pDiag)
{
   EnterCriticalSection(&csDllMain);
   __try
   {
      pTPCC = (TPCC_STATE *) calloc(1,sizeof(TPCC_STATE));
      if (pTPCC == NULL)
      {
         wsprintf(pDiag,"ThrAtt(%ld): pTPCC Alloc Failed (%ld)\n",
            GetCurrentThreadId(),GetLastError);
         DiagIoWrite(pDiag,DIAG_ERROR);
         return(TRUE);
      };
      TlsSetValue(dwTlsInx,pTPCC);
      pTPCC->tsTMon.pszErrTxt = pTPCC->ErrTxt;
      if (TMInit(&pTPCC->tsTMon))
      {
         wsprintf(pDiag,"ThrAtt(%ld): TMInit %s\n",
            GetCurrentThreadId(),pTPCC->ErrTxt);
         DiagIoWrite(pDiag,DIAG_ERROR);
         return(TRUE);
      };
   }
   __finally
   {
      LeaveCriticalSection(&csDllMain);
   };
   return(FALSE);
};   // ThreadAttach

//============================================================================
//
// Function name: ThreadDetach
//
//============================================================================
VOID ThreadDetach(TPCC_STATE * pTPCC)
{
   EnterCriticalSection(&csDllMain);
   __try
   {
      pTPCC = (TPCC_STATE *) TlsGetValue(dwTlsInx);
      if (pTPCC != NULL)
      {
         TMDone(&pTPCC->tsTMon);
         free(pTPCC);
         pTPCC = NULL;
         TlsSetValue(dwTlsInx,pTPCC);
      };
   }
   __finally
   {
      LeaveCriticalSection(&csDllMain);
   };
};   // ThreadDetach

//============================================================================
//
// Function name: GetExtensionVersion
//
//============================================================================
BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO *pVersion)
{
   pVersion->dwExtensionVersion = EXTN_VERSION;
   strncpy(pVersion->lpszExtensionDesc,pTitle,HSE_MAX_EXT_DLL_NAME_LEN);
   return TRUE;
};   // GetExtensionVersion

//============================================================================
//
// Function name: HttpExtensionProc
//
//
// Returns:
//    HSE_STATUS_SUCCESS               send msg, drop connection
//    HSE_STATUS_SUCCESS_AND_KEEP_CONN       send msg, keep connection
//
//============================================================================
DWORD WINAPI HttpExtensionProc(EXTENSION_CONTROL_BLOCK * pECB)
{
   TPCC_STATE * pTPCC;
   DWORD dwRslt = HSE_STATUS_SUCCESS;
   UINT uRslt;

   pTPCC = (TPCC_STATE *) TlsGetValue(dwTlsInx);
   if (pTPCC == NULL)
   {
      CHAR szWork[200];
      ThreadAttach(pTPCC,szWork);
      pTPCC = (TPCC_STATE *) TlsGetValue(dwTlsInx);
      if (pTPCC == NULL)
      {
         SendResponse(pECB,szTPCCError,szWork);
         goto HttpXit;
      };
   };
   if (pTPCC->tsTMon.pTxnData == NULL)
      SendResponse(pECB,szTMInitError,pTPCC->szHeader);
   TPCCClear(pTPCC);
   pTPCC->ConnID = pECB->ConnID;
   pTPCC->RecvMsg = pECB->lpszQueryString;
   uRslt = TPCCHandler(pTPCC);
   switch (uRslt)
   {
      case TPCCSEND:
         SendResponse(pECB,pTPCC->SendMsg,pTPCC->szHeader);
         dwRslt = HSE_STATUS_SUCCESS_AND_KEEP_CONN;
         break;
```

```
      case TPCCSENDEND:
         SendResponse(pECB,pTPCC->SendMsg,pTPCC->szHeader);
         break;
      case TPCCENDNOW:
      default:
         break;
   };   // switch (TPCCHandler result)

HttpXit:

       return(dwRslt);

};   // HttpExtensionProc

//====================================================================
//
// Function name: SendResponse
//
//====================================================================
VOID SendResponse(EXTENSION_CONTROL_BLOCK * pECB,CHAR * pMsg,CHAR * pWork)
{
   DWORD dwMsgBytes;
   DWORD dwDataBytes;
   CHAR * pCL;
   HSE_SEND_HEADER_EX_INFO HeaderExInfo;
   dwMsgBytes = strlen(pMsg);
   pCL=strstr(pMsg,CTEXT);
   dwDataBytes = dwMsgBytes - iHHdrLen;
   wsprintf(pWork,"%4ld",dwDataBytes);
   pCL += iCTextLen;
   strncpy(pCL,pWork,4);
   HeaderExInfo.pszHeader = pMsg;
   HeaderExInfo.cchHeader = dwMsgBytes;
   HeaderExInfo.pszStatus = "200 OK";
   HeaderExInfo.cchStatus = 6;
   HeaderExInfo.fKeepConn = TRUE;
   (*pECB->ServerSupportFunction)
      (pECB->ConnID,
      HSE_REQ_SEND_RESPONSE_HEADER_EX,
      &HeaderExInfo,
      NULL,
      NULL);
};   // SendResponse

//====================================================================
//
// Function name: ReadRegistry
//
//    Sets global operational parameters from registry if they exist.
//    Otherwise, compiled in defaults apply.
//
// Result:
//    FALSE    Registry entry found
//    TRUE     Registry entry does not exist
//
//====================================================================
BOOL ReadRegistry(VOID)
{
   HKEY hkTPCC;
   DWORD       dwMax;
   DWORD       dwRT;
   INT i;
   CHAR szValue[100];
   if (RegOpenKeyEx(HKEY_LOCAL_MACHINE,"SOFTWARE\\Unisys\\TPCC",0,
      KEY_READ, &hkTPCC) != ERROR_SUCCESS )
      return(TRUE);
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"EVENTLOG",0,&dwRT,(BYTE *) &szValue,&dwMax)
      == ERROR_SUCCESS)
   {
      if (abs(atoi(szValue) == 0))
         bSetEventLog = FALSE;
      else
         bSetEventLog = TRUE;
   };
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"CONSOLE",0,&dwRT,(BYTE *) &szValue,&dwMax)
      == ERROR_SUCCESS )
   {
      if (abs(atoi(szValue) == 0))
         bSetConsole = FALSE;
      else
         bSetConsole = TRUE;
   };
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"DIAGLEVEL",0,&dwRT,(BYTE *)
&szValue,&dwMax)
      == ERROR_SUCCESS )
   {
      i = atoi(szValue);
      if (i < DIAG_FORCE)
         i = DIAG_FORCE;
      else
      if (i > DIAG_INFO)
         i = DIAG_INFO;
      uSetDiagLevel = i;
   };
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"MAXTERMS",0,&dwRT,(BYTE *) &szValue,&dwMax)
      == ERROR_SUCCESS )
   {
      iMaxTerms = abs(atoi(szValue));
   };
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"DELIVERYTHREADS",0,&dwRT,(BYTE *)
&szValue,&dwMax)
      == ERROR_SUCCESS )
   {
      lSetDThreads = abs(atoi(szValue));
   };
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"DQSIZE",0,&dwRT,(BYTE *) &szValue,&dwMax)
      == ERROR_SUCCESS )
   {
      lSetDQSize = abs(atoi(szValue));
   };
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"DQPATH",0,&dwRT,(BYTE *) &szValue,&dwMax)
      == ERROR_SUCCESS )
   {
      strcpy(szSetPath,szValue);
   };
```

```
   RegCloseKey(hkTPCC);
   return(FALSE);
};   // ReadRegistry
```

tpcchandler.h

```
// tpcchandler.h
//
// Copyright Unisys, 1999

typedef struct
{
   LPVOID ConnID;                    // Active Connection Id
   SHORT sWId;                       // TPCC WareHouse Id
   SHORT sDId;                       // TPCC District Id
   INT  iSyncId;                     // TPCC Sync Id
   INT  iTermId;                     // TPCC Term Id
   UINT uFormId;                     // TPCC Form Id
   INT  iStatusId;                   // TPCC Status Id
   CHAR ErrTxt[500];                 // Error text
   CHAR szWork[200];                 // Thread work area
   CHAR szHeader[100];               // HTTP work area
   CHAR * RecvMsg;                   // HTML message from ECB
   CHAR SendMsg[MAX_MSG_SZ];         // HTML work area
   TMON_STATE tsTMon;                // TMon Interface
} TPCC_STATE;

BOOL TPCCClear(TPCC_STATE * pTPCC);
UINT TPCCHandler(TPCC_STATE * pTPCC);
```

tpcchandler.cpp

```
// tpcchandler.cpp
//
// Copyright Unisys, 1999
//
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "..\tpccsvr\tpcc.h"
#include "tmon.h"
#include "diagio.h"
#include "delivery.h"
#include "tpcchandler.h"
#include "term.h"

// pTPCC->iFormId - TPCC forms enumeration.
#define FORM_NULL        0
#define FORM_LOGON       1
#define FORM_MENU        2
#define FORM_NEWORDER    3
#define FORM_PAYMENT     4
#define FORM_DELIVERY    5
#define FORM_ORDERSTATUS 6
#define FORM_STOCKLEVEL  7
#define FORM_EXIT        8
#define FORM_MAX         9

// CMD= HTML Command Enumeration and Name
#define CMD_NULL              0
#define CMD_PROCESS           1
#define CMD_NEWORDER_FORM     2
#define CMD_PAYMENT_FORM      3
#define CMD_DELIVERY_FORM     4
#define CMD_ORDERSTATUS_FORM  5
#define CMD_STOCKLEVEL_FORM   6
#define CMD_EXIT              7
#define CMD_SUBMIT            8
#define CMD_MENU_FORM         9
#define CMD_MAX               10

static CHAR * szCmds[] =
{
   "Unknown",
   "Process",
   "..NewOrder..",
   "..Payment..",
   "..Delivery..",
   "..Order-Status..",
   "..Stock-Level..",
   "..Exit..",
   "Submit",
   "Menu"
};

static CHAR * szFormLogin =
   HTTPHdr "<HTML>"
   "<HEAD><TITLE>Welcome To TPC-C</TITLE></HEAD><BODY>"
   "Please Identify your Warehouse and District for this session.<BR>"
   "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
   "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"
   "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"1\">"
   "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"-2\">"
   "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"0\">"
   "Warehouse ID <INPUT NAME=\"w_id\" SIZE=4><BR>"
   "District ID <INPUT NAME=\"d_id\" SIZE=2><BR>"
   "<HR>"
   "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Submit\">"
   "</FORM>";

static CHAR * szMenuList =
   "<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..\">";

static CHAR * HTMLTrailer =
   "</BODY></HTML>";

static CHAR * TERMIDTOKEN = "TERMID=";
static CHAR * SYNCIDTOKEN = "SYNCID=";
static CHAR * FORMIDTOKEN = "FORMID=";
static CHAR * STATUSIDTOKEN = "STATUSID=";
static CHAR * CMDTOKEN = "CMD=";
```

```
static CHAR * NEWORDER_SERVICE = "NEWORDER";
static CHAR * PAYMENT_SERVICE = "PAYMENT";
static CHAR * ORDERSTATUS_SERVICE = "ORDERSTS";
static CHAR * DELIVERY_SERVICE = "DELIVERY";
static CHAR * STOCKLEVEL_SERVICE = "STOCKLVL";
static CHAR * ZIPPIC = "XXXXX-XXXX";

BOOL ProcessLogin(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC);
BOOL ProcessForm(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC);
BOOL ProcessNewOrder(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC);
BOOL ProcessPayment(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC);
BOOL ProcessDelivery(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC);
BOOL ProcessOrderStatus(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC);
BOOL ProcessStockLevel(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC);
VOID FormatLogin(CHAR * pMsg,CHAR * pAddText);
BOOL GetHidden(CHAR * pMsg,UINT * uFormId,INT * iSyncId,INT * iTermId);
BOOL GetCmd(CHAR * pMsg,CHAR * pWork,UINT uLen);
BOOL GetLongKey(LONG * lRslt,CHAR * pHTML,CHAR * pKey,TPCC_STATE * pTPCC);
BOOL GetIntKey(INT * iRslt,CHAR * pHTML,CHAR * pKey,TPCC_STATE * pTPCC);
BOOL GetShortKey(SHORT * sRslt,CHAR * pHTML,CHAR * pKey,TPCC_STATE *
pTPCC);
BOOL GetStringKey(CHAR * szRslt,CHAR * pHTML,CHAR * pKey,
                  TPCC_STATE * pTPCC,UINT uMax);
BOOL GetAmountKey(DOUBLE * dRslt,CHAR * pHTML,CHAR * pKey,
                  TPCC_STATE * pTPCC);
BOOL GetKeyValue(CHAR * pHTML,CHAR * pKey,CHAR * pValue,UINT uMax);
VOID FormatLogin(CHAR * pOut,CHAR * pAddText);
VOID FormatMenu(CHAR * pOut,TPCC_STATE * pTPCC);
VOID FormatNewOrder(CHAR * pOut,TPCC_STATE * pTPCC);
VOID FormatPayment(CHAR * pOut,TPCC_STATE * pTPCC);
VOID FormatDelivery(CHAR * pOut,TPCC_STATE * pTPCC);
VOID FormatOrderStatus(CHAR * pOut,TPCC_STATE * pTPCC);
VOID FormatStockLevel(CHAR * pOut,TPCC_STATE * pTPCC);
INT FormatFormHdr(CHAR * pOut,CHAR * pTitle,TPCC_STATE * pTPCC);
INT FormatRespHdr(CHAR * pOut,CHAR * pTitle,TPCC_STATE * pTPCC);
VOID FormatString(CHAR * pOut,CHAR * pPic,CHAR * pIn);
VOID UtilStrCpy(CHAR * pDest,CHAR * pSrc,INT n);
BOOL CheckNumeric(CHAR * pNum);

//============================================================================
//
// Function name: TPCCClear
//
//============================================================================
BOOL TPCCClear(TPCC_STATE * pTPCC)
{
   pTPCC->ConnID = 0;
   pTPCC->sWId = 0;
   pTPCC->sDId = 0;
   pTPCC->iSyncId = 0;
   pTPCC->iTermId = -2;
   pTPCC->uFormId = FORM_NULL;
   pTPCC->iStatusId = 0;
   strcpy(pTPCC->ErrTxt,"");
   return(FALSE);
};   // TPCCClear

//============================================================================
//
// Function name: TPCCHandler
//
//============================================================================
UINT TPCCHandler(TPCC_STATE * pTPCC)
{
   INT iSyncId;
   INT iTermId;
   UINT uCmdId;
   UINT uRslt = TPCCSENDEND;  // default error handling
   TERM_STATE * pTerm;

   pTPCC->iStatusId = STATUS_OK;
   if (GetHidden(pTPCC->RecvMsg,&pTPCC->uFormId,&iSyncId,&iTermId))
   {
      uRslt = TPCCSEND;
      FormatLogin(pTPCC->SendMsg,pTPCC->ErrTxt);
      goto HdlrXit;
   };
   if (iTermId > 0)
   {
      pTerm = TermGet(iTermId);
      if (pTerm == NULL)
      {
         uRslt = TPCCSEND;
         strcpy(pTPCC->ErrTxt,"Invalid Term Id");
         FormatLogin(pTPCC->SendMsg,pTPCC->ErrTxt);
         goto HdlrXit;
      };
      pTPCC->sWId = pTerm->sWId;
      pTPCC->sDId = pTerm->sDId;
      pTPCC->iSyncId = pTerm->iSyncId;
      pTPCC->iTermId = pTerm->iTermId;
   };
   uCmdId = GetCmd(pTPCC->RecvMsg,pTPCC->szWork,sizeof(pTPCC->szWork));
   // Except for Submit(log in), sWId must already be set
   if (pTPCC->sWId == 0 && uCmdId != CMD_SUBMIT)
   {
      strcpy(pTPCC->ErrTxt,"Must log in first!");
      FormatLogin(pTPCC->SendMsg,pTPCC->ErrTxt);
      uRslt = TPCCSEND;
      goto HdlrXit;
   };
   // Check for multiple log in attempts
   if (pTPCC->sWId != 0 && uCmdId == CMD_SUBMIT)
   {
      strcpy(pTPCC->ErrTxt,ERRTXT_ALREADY_LOGGEDIN);
      pTPCC->iStatusId = ERR_ALREADY_LOGGEDIN;
      FormatMenu(pTPCC->SendMsg,pTPCC);
      uRslt = TPCCSEND;
      goto HdlrXit;
   };
   // If not logging in, validate hidden fields
   if (uCmdId != CMD_SUBMIT)
   {
      if (iTermId != pTPCC->iTermId || iTermId != iSyncId)
      {
         wsprintf(pTPCC->ErrTxt,"%s: Received %ld, %ld (%ld)",
            ERRTXT_TERMID,iTermId,iSyncId,pTPCC->iTermId);
         pTPCC->iStatusId = ERR_TERMID;
         FormatMenu(pTPCC->SendMsg,pTPCC);
         goto HdlrXit;
      };
   };
```

```
   // Process the command
   switch (uCmdId)
   {
      case CMD_SUBMIT:
         ProcessLogin(pTPCC->RecvMsg,pTPCC->SendMsg,pTPCC);
         break;
      case CMD_MENU_FORM:
         FormatMenu(pTPCC->SendMsg,pTPCC);
         break;
      case CMD_PROCESS:
         ProcessForm(pTPCC->RecvMsg,pTPCC->SendMsg,pTPCC);
         break;
      case CMD_NEWORDER_FORM:
         FormatNewOrder(pTPCC->SendMsg,pTPCC);
         break;
      case CMD_PAYMENT_FORM:
         FormatPayment(pTPCC->SendMsg,pTPCC);
         break;
      case CMD_DELIVERY_FORM:
         FormatDelivery(pTPCC->SendMsg,pTPCC);
         break;
      case CMD_ORDERSTATUS_FORM:
         FormatOrderStatus(pTPCC->SendMsg,pTPCC);
         break;
      case CMD_STOCKLEVEL_FORM:
         FormatStockLevel(pTPCC->SendMsg,pTPCC);
         break;
      case CMD_EXIT:
         TermFree(pTPCC->iTermId);
         strcpy(pTPCC->ErrTxt,"Logged Off");
         FormatLogin(pTPCC->SendMsg,pTPCC->ErrTxt);
         goto HdlrXit;
      default:
         strcpy(pTPCC->ErrTxt,ERRTXT_CMD_UNKNOWN);
         pTPCC->iStatusId = ERR_CMD_UNKNOWN;
         if (pTPCC->sWId == 0)
            FormatLogin(pTPCC->SendMsg,pTPCC->ErrTxt);
         else
            FormatMenu(pTPCC->SendMsg,pTPCC);
         break;
   };   // switch (uCmdId)

   uRslt = TPCCSEND;

HdlrXit:

   return(uRslt);

};   // TPCCHandler

//================================================================================
//
// Function name: ProcessLogin
//
//    ProcessLogin extracts WId and DId from the incoming form.  Assumes
//    log in has not previously completed (sWId == 0 already verified).
//
//    Result:
//       FALSE - log in successful, sWId and sDId set in pTPCC,
//               pOut contains menu.
//       TRUE - log in failed, pOut contains log in form with
//              error message.
//
//================================================================================
BOOL ProcessLogin(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC)
{
   SHORT sWId;
   SHORT sDId;
   TERM_STATE * pTerm;

   if (GetShortKey(&sWId,pIn,"w_id",pTPCC))
   {
     FormatLogin(pOut,pTPCC->ErrTxt);
     return(TRUE);
   };
   if (sWId < 1)
   {
      wsprintf(pTPCC->ErrTxt,"Warehouse Id (%d) Invalid",sWId);
      pTPCC->iStatusId = ERR_WID_INVALID;
      FormatLogin(pOut,pTPCC->ErrTxt);
      return(TRUE);
   };
   if (GetShortKey(&sDId,pIn,"d_id",pTPCC))
   {
     FormatLogin(pOut,pTPCC->ErrTxt);
     return(TRUE);
   };
   if (sDId < MIN_DId || sDId > MAX_DId)
   {
      wsprintf(pTPCC->ErrTxt,"DId Out of Range(%ld,%ld) - %ld",
         MIN_DId,MAX_DId,sDId);
      pTPCC->iStatusId = ERR_DID_INVALID;
      FormatLogin(pOut,pTPCC->ErrTxt);
      return(TRUE);
   };
   pTerm = TermAlloc();
   if (pTerm == NULL)
   {
      wsprintf(pTPCC->ErrTxt,"Unable to Allocate Terminal Entry");
      pTPCC->iStatusId = ERR_TERM_ALLOC;
      FormatLogin(pOut,pTPCC->ErrTxt);
      return(TRUE);
   };
   pTerm->ConnID = pTPCC->ConnID;
   pTerm->iSyncId = pTerm->iTermId;
   pTerm->sWId = abs(sWId);
   pTerm->sDId = abs(sDId);
   pTPCC->iTermId = pTerm->iTermId;
   pTPCC->iSyncId = pTerm->iSyncId;
   pTPCC->sWId = pTerm->sWId;
   pTPCC->sDId = pTerm->sDId;
   FormatMenu(pOut,pTPCC);
   return(FALSE);
};   // ProcessLogin

//================================================================================
//
// Function name: ProcessForm
//
//    ProcessForm uses pTPCC->uFormId to determine which form input is
//    present and ready for processing.  Actual processing is done by
//    the form specific routine.
```

```
//
//    Result:
//       FALSE - form processed, pOut contains response.
//       TRUE - error processing form input, pOut contains reason.
//
//========================================================================
BOOL ProcessForm(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC)
{
   switch (pTPCC->uFormId )
   {
      case FORM_NEWORDER:
               return(ProcessNewOrder(pIn,pOut,pTPCC));
      case FORM_PAYMENT:
               return(ProcessPayment(pIn,pOut,pTPCC));
      case FORM_DELIVERY:
               return(ProcessDelivery(pIn,pOut,pTPCC));
      case FORM_ORDERSTATUS:
               return(ProcessOrderStatus(pIn,pOut,pTPCC));
      case FORM_STOCKLEVEL:
               return(ProcessStockLevel(pIn,pOut,pTPCC));
      default:
         wsprintf(pTPCC->ErrTxt,"%s (%ld)",
            ERRTXT_FORM_UNKNOWN,pTPCC->uFormId);
         pTPCC->iStatusId = ERR_FORM_UNKNOWN;
         FormatMenu(pOut,pTPCC);
         break;
      }
   return(TRUE);
};   // ProcessForm

//========================================================================
//
// Function name: ProcessNewOrder
//
//    ProcessNewOrder extracts the input data fields from pIn, processes
//    the data, and returns a response in pOut.
//
//    Result:
//       FALSE - NewOrder processed successfully.
//       TRUE - NewOrder processing failed.
//
//========================================================================
BOOL ProcessNewOrder(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC)
{
   NEW_ORDER_DATA * pnod;
   TMON_STATE * pTMon;
   CHAR szKey[20];
   CHAR * ptr;
   INT iInx;
   UINT u;
   BOOL bDone = FALSE;
   HRESULT hr;
   int iSize;

   pTMon = &pTPCC->tsTMon;
   pnod = (NEW_ORDER_DATA *) pTMon->pTxnData;
   ZeroMemory(pnod,sizeof(NEW_ORDER_DATA));
   pnod->w_id = pTPCC->sWId;
   if (GetShortKey(&pnod->d_id,pIn,"DID*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (pnod->d_id < MIN_DId || pnod->d_id > MAX_DId)
   {
      wsprintf(pTPCC->ErrTxt,"DId Out of Range(%ld,%ld) - %ld",
         MIN_DId,MAX_DId,pnod->d_id);
      pTPCC->iStatusId = ERR_DID_INVALID;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   if (GetLongKey(&pnod->c_id,pIn,"CID*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   pnod->o_ol_cnt = 0;
   ptr = pIn;
   for(u=0; u < MAX_OL; u++)
   {
      wsprintf(szKey,"SP%2.2d*",u);
      ptr = strstr(ptr,szKey);
      if (GetShortKey(&pnod->Ol[u].ol_supply_w_id,ptr,szKey,pTPCC))
      {
         FormatMenu(pOut,pTPCC);
         return(TRUE);
      };
      wsprintf(szKey,"IID%2.2d*",u);
      if (GetLongKey(&pnod->Ol[u].ol_i_id,ptr,szKey,pTPCC))
      {
         FormatMenu(pOut,pTPCC);
         return(TRUE);
      };
      wsprintf(szKey,"Qty%2.2d*",u);
      if (GetShortKey(&pnod->Ol[u].ol_quantity,ptr,szKey,pTPCC))
      {
         FormatMenu(pOut,pTPCC);
         return(TRUE);
      };
      if (pnod->Ol[u].ol_i_id != 0)
      {
         // Check for prior blank lines
         if (bDone)
         {
            strcat(pTPCC->ErrTxt,"Embedded Empty Order Lines");
            pTPCC->iStatusId = ERR_EMBEDDED_EMPTY_OL;
            FormatMenu(pOut,pTPCC);
            return(TRUE);
         };
         if (pnod->Ol[u].ol_supply_w_id < 1)
         {
            wsprintf(pTPCC->ErrTxt,
               "Order Line %ld Contains Invalid WId %d",
               u,pnod->Ol[u].ol_supply_w_id);
            pTPCC->iStatusId = ERR_WID_INVALID;
            FormatMenu(pOut,pTPCC);
            return(TRUE);
         };
         if (pnod->Ol[u].ol_quantity < MIN_QUANTITY ||
             pnod->Ol[u].ol_quantity > MAX_QUANTITY)
         {
            wsprintf(pTPCC->ErrTxt,
```

```
               "Order Line %ld Contains Invalid Qty %d",
               u,pnod->Ol[u].ol_quantity);
            pTPCC->iStatusId = ERR_QUANTITY_INVALID;
            FormatMenu(pOut,pTPCC);
            return(TRUE);
         };
         pnod->o_ol_cnt++;
      }   // if (ol_i_id !=0)
      else
      {
         if (pnod->Ol[u].ol_supply_w_id != 0)
         {
            wsprintf(pTPCC->ErrTxt,
               "Order Line %ld WId Supplied with No Item",u);
            pTPCC->iStatusId = ERR_OL_INVALID;
            FormatMenu(pOut,pTPCC);
            return(TRUE);
         };
         if (pnod->Ol[u].ol_quantity != 0)
         {
            wsprintf(pTPCC->ErrTxt,
               "Order Line %ld Qty Supplied with No Item",u);
            pTPCC->iStatusId = ERR_OL_INVALID;
            FormatMenu(pOut,pTPCC);
            return(TRUE);
         };
         bDone = TRUE;
      };   // empty order line
   };   // for (u < MAX_OL)

   if (pnod->o_ol_cnt < MIN_OL)
   {
      wsprintf(pTPCC->ErrTxt,"Too Few Order Lines %d",pnod->o_ol_cnt);
      pTPCC->iStatusId = ERR_OL_COUNT;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   iSize = pTMon->iSize;
   hr = pTMon->pIAllTxn->NewOrder(&iSize,(unsigned char**)&pTMon-
>pTxnData);
   if (FAILED(hr))
   {
      pTPCC->iStatusId = ERR_TM_INTERFACE;
      wsprintf(pTPCC->ErrTxt,
         "COM Interface to NewOrder Call Failed, HRESULT %x",
         hr);
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   pnod = (NEW_ORDER_DATA *) pTMon->pTxnData;
   // Exclude invalid item id case
   if (pnod->bTPRslt && pnod->iTPRslt < SVC_NOERROR)
   {
      wsprintf(pTPCC->ErrTxt,
         "New Order Service Returned Error(%ld): %s",
         pnod->iTPRslt,pnod->execution_status);
      pTPCC->iStatusId = ERR_SERVICE_RSLT;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   if (pnod->iTPRslt == SVC_BADITEMID)
      pTPCC->iStatusId = INVALID_IID;

   iInx = FormatRespHdr(pOut,"TPC-C New Order",pTPCC);
   if (!pnod->bTPRslt)
   {
      iInx += wsprintf(pOut + iInx,
         "<PRE>                                New Order<BR>"
         "Warehouse: %4.4d   District: %2.2d                          "
         "Date: %2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d <BR>"
         "Customer:  %4.4d   Name: %-16s   Credit: %-2s   ",
         pnod->w_id,pnod->d_id,
         pnod->o_entry_d.day,pnod->o_entry_d.month,
         pnod->o_entry_d.year,pnod->o_entry_d.hour,
         pnod->o_entry_d.minute,pnod->o_entry_d.second,
         pnod->c_id,pnod->c_last,pnod->c_credit);
      iInx += sprintf(pOut + iInx,
         "%%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>",
         pnod->c_discount * 100,pnod->o_id,pnod->o_ol_cnt,pnod->w_tax *
100,pnod->d_tax * 100);
      for (u = 0; u < (UINT) pnod->o_ol_cnt; u++)
      {
         iInx += sprintf(pOut + iInx,
            "  %4.4d   %6.6d   %-24s  %2.2d    %3.3d    %1.1s   $%6.2f
$%7.2f  <BR>",
            pnod->Ol[u].ol_supply_w_id,pnod->Ol[u].ol_i_id,
            pnod->Ol[u].ol_i_name,pnod->Ol[u].ol_quantity,pnod-
>Ol[u].ol_stock,
            pnod->Ol[u].ol_brand_generic,pnod->Ol[u].ol_i_price,
            pnod->Ol[u].ol_amount);
      };
      for(; u < MAX_OL; u++)
      {
         strcat(pOut + iInx," <BR>");
         iInx += 5;
      };
      sprintf(pOut + iInx,
         "Execution Status: %24.24s                  Total:  $%8.2f  "
         "</PRE><HR><BR>%s</FORM>%s",
         pnod->execution_status,pnod->total_amount,
         szMenuList,HTMLTrailer);
   }   // !bTPRslt
   else
   {
      iInx += wsprintf(pOut + iInx,
         "<PRE>                                New Order<BR>"
         "Warehouse: %4.4d   District: %2.2d
Date:<BR>"
         "Customer:  %4.4d   Name: %-16s   Credit: %-2s   "
         "%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>",
         pnod->w_id,pnod->d_id,
         pnod->c_id,pnod->c_last,pnod->c_credit,
         pnod->o_id);
      for(u = 0; u < MAX_OL; u++)
```

```
      {
         strcat(pOut + iInx," <BR>");
         iInx += 5;
      };
      iInx += wsprintf(pOut + iInx,
         "Execution Status: %24.24s                    Total:"
         "</PRE><HR><BR>%s</FORM>%s",
         pnod->execution_status,szMenuList,HTMLTrailer);
   };   // bTPRslt

   return(FALSE);

};   // ProcessNewOrder

//=========================================================================
//
// Function name: ProcessPayment
//
//    ProcessPayment extracts the input data fields from pIn, processes
//    the data, and returns a response in pOut.
//
//    Result:
//       FALSE - Payment processed successfully.
//       TRUE - Payment processing failed.
//
//=========================================================================
BOOL ProcessPayment(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC)
{
   PAYMENT_DATA * ppd;
   TMON_STATE * pTMon;
   CHAR szWork1[60];
   CHAR szZip1[20];
   CHAR szZip2[20];
   CHAR szZip3[20];
   INT iInx;
   HRESULT hr;
   int iSize;

   pTMon = &pTPCC->tsTMon;
   ppd = (PAYMENT_DATA *) pTMon->pTxnData;
   ZeroMemory(ppd,sizeof(PAYMENT_DATA));
   ppd->w_id = pTPCC->sWId;
   // Get and validate DId
   if (GetShortKey(&ppd->d_id,pIn,"DID*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (ppd->d_id < MIN_DId || ppd->d_id > MAX_DId)
   {
      wsprintf(pTPCC->ErrTxt,"DId Out of Range(%ld,%ld) - %ld",
         MIN_DId,MAX_DId,ppd->d_id);
      pTPCC->iStatusId = ERR_DID_INVALID;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   // Get and validate customer Id and name
   if (GetLongKey(&ppd->c_id,pIn,"CID*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (GetStringKey(ppd->c_last,pIn,"CLT*",pTPCC,NAME_LEN))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (ppd->c_id == 0 && ppd->c_last[0] == 0)
   {
      strcpy(pTPCC->ErrTxt,"Error - Customer Id and Name Empty");
      pTPCC->iStatusId = ERR_IDANDNAME_EMPTY;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   if (ppd->c_id != 0 && ppd->c_last[0] != 0)
   {
      strcpy(pTPCC->ErrTxt,
         "Error - Specify Customer Id or Name, not Both");
      pTPCC->iStatusId = ERR_IDANDNAME_ENTERED;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   // Get and validate customer DId
   if (GetShortKey(&ppd->c_d_id,pIn,"CDI*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (ppd->c_d_id < MIN_DId || ppd->c_d_id > MAX_DId)
   {
      wsprintf(pTPCC->ErrTxt,"Cust DId Out of Range(%ld,%ld) - %ld",
         MIN_DId,MAX_DId,ppd->d_id);
      pTPCC->iStatusId = ERR_DID_INVALID;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   // Get and validate customer WId
   if (GetShortKey(&ppd->c_w_id,pIn,"CWI*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (ppd->c_w_id < 1)
   {
      wsprintf(pTPCC->ErrTxt,
         "Payment Contains Invalid Customer WId %d",
         ppd->c_w_id);
      pTPCC->iStatusId = ERR_WID_INVALID;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   // Get and validate amount
   if (GetAmountKey(&ppd->h_amount,pIn,"HAM*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (ppd->h_amount <= 0)
   {
      wsprintf(pTPCC->ErrTxt,
         "Payment Amount Negative or Missing");
      pTPCC->iStatusId = ERR_AMOUNT_INVALID;
```

```
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   iSize = pTMon->iSize;
   hr = pTMon->pIAllTxn->Payment(&iSize,(unsigned char**)&pTMon-
>pTxnData);
   if (FAILED(hr))
   {
      pTPCC->iStatusId = ERR_TM_INTERFACE;
      wsprintf(pTPCC->ErrTxt,
         "COM Interface to Payment Call Failed, HRESULT %x",
         hr);
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   ppd = (PAYMENT_DATA *) pTMon->pTxnData;
   if (ppd->bTPRslt)
   {
      wsprintf(pTPCC->ErrTxt,
         "Payment Service Returned Error(%ld): %s",
         ppd->iTPRslt,ppd->execution_status);
      pTPCC->iStatusId = ERR_SERVICE_RSLT;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   iInx = FormatRespHdr(pOut,"TPC-C Payment",pTPCC);
   FormatString(szZip1,ZIPPIC,ppd->w_zip);
   FormatString(szZip2,ZIPPIC,ppd->d_zip);
   FormatString(szZip3,ZIPPIC,ppd->c_zip);
   FormatString(szWork1,"XXXXXX-XXX-XXX-XXXX",ppd->c_phone);
   iInx += wsprintf(pOut + iInx,
          "<PRE>                                      Payment<BR>"
      "Date: %2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d <BR><BR>"
      "Warehouse: %4.4d"
      "                          District: %2.2d<BR>"
      "%-20s                     %-20s<BR>"
      "%-20s                     %-20s<BR>"
      "%-20s %-2s %10.10s       %-20s %-2s %10.10s<BR><BR>"
      "Customer: %4.4d  Cust-Warehouse: %4.4d  Cust-District: %2.2d<BR>"
      "Name:   %-20s %-2s %-20s     Since:  %2.2d-%2.2d-%4.4d<BR>"
      "        %-20s                     Credit: %-2s<BR>"
      "        %-20s                     ",
          ppd->h_date.day,ppd->h_date.month,
      ppd->h_date.year,ppd->h_date.hour,
      ppd->h_date.minute,ppd->h_date.second,
      ppd->w_id,ppd->d_id,
      ppd->w_street_1,ppd->d_street_1,
      ppd->w_street_2,ppd->d_street_2,
      ppd->w_city,ppd->w_state,szZip1,ppd->d_city,ppd->d_state,szZip2,
      ppd->c_id,ppd->c_w_id,ppd->c_d_id,
      ppd->c_first,ppd->c_middle,ppd->c_last,
      ppd->c_since.day,ppd->c_since.month,ppd->c_since.year,
      ppd->c_street_1,ppd->c_credit,ppd->c_street_2);
   iInx += sprintf(pOut + iInx,"%%Disc:  %5.2f<BR>",ppd->c_discount *
100);
   iInx += wsprintf(pOut + iInx,
      "        %-20s %-2s %10.10s       Phone:  %-19.19s<BR><BR>",
      ppd->c_city,ppd->c_state,szZip3,szWork1);
   iInx += sprintf(pOut + iInx,
      "Amount Paid:          $%7.2f      New Cust Balance: $%14.2f<BR>"
      "Credit Limit:   $%13.2f<BR><BR>",
      ppd->h_amount,ppd->c_balance,ppd->c_credit_lim);
   if (ppd->c_credit[0] == 'B' && ppd->c_credit[1] == 'C')
   {
      wsprintf(pOut + iInx,
         "Cust-Data: %-50.50s<BR>           %-50.50s<BR>           "
         "%-50.50s<BR>           %-50.50s<BR>"
         "</PRE><HR><BR>%s</FORM>%s",
         ppd->c_data,(ppd->c_data + 50),(ppd->c_data + 100),(ppd->c_data +
150),
         szMenuList,HTMLTrailer);
   }
   else
   {
      wsprintf(pOut + iInx,
         "Cust-Data: <BR><BR><BR><BR>"
         "</PRE><HR><BR>%s</FORM>%s",
         szMenuList,HTMLTrailer);
   };

   return(FALSE);

};   // ProcessPayment

//============================================================================
//
// Function name: ProcessDelivery
//
//    ProcessDelivery extracts the input data fields from pIn, processes
//    the data, and returns a response in pOut.
//
//    Result:
//       FALSE - Delivery processed successfully.
//       TRUE - Delivery processing failed.
//
//============================================================================
BOOL ProcessDelivery(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC)
{
   DELIVERY_DATA * pdd;
   TMON_STATE * pTMon;
   INT iInx;

   pTMon = &pTPCC->tsTMon;
   pdd = (DELIVERY_DATA *) pTMon->pTxnData;
   ZeroMemory(pdd,sizeof(DELIVERY_DATA));
   pdd->w_id = pTPCC->sWId;
   // Get and validate carrier id
   if (GetShortKey(&pdd->o_carrier_id,pIn,"OCD*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (pdd->o_carrier_id < MIN_CARRIER ||
       pdd->o_carrier_id > MAX_CARRIER)
   {
      wsprintf(pTPCC->ErrTxt,"Carrier Id Out of Range(%ld,%ld) - %ld",
         MIN_CARRIER,MAX_CARRIER,pdd->o_carrier_id);
      pTPCC->iStatusId = ERR_CARRIER_INVALID;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   GetLocalTime(&pdd->QTime);
```

```
   DeliveryPost(pdd);
   if (pdd->bTPRslt)
   {
      wsprintf(pTPCC->ErrTxt,
         "Delivery Post Returned Error(%ld): Queue Request Failed",
         pdd->iTPRslt);
      pTPCC->iStatusId = ERR_SERVICE_RSLT;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   iInx = FormatRespHdr(pOut,"TPC-C Delivery",pTPCC);
   iInx += wsprintf(pOut + iInx,
      "<PRE>                                    Delivery<BR>"
      "Warehouse: %4.4d<BR><BR>"
      "Carrier Number: %2.2d<BR><BR>"
      "Execution Status: %25.25s<BR>"
      "</PRE><HR><BR>%s</FORM>%s",
      pdd->w_id,pdd->o_carrier_id,"Delivery has been queued.",
      szMenuList,HTMLTrailer);

   return(FALSE);

};   // ProcessDelivery

//================================================================
//
// Function name: ProcessOrderStatus
//
//    ProcessOrderStatus extracts the input data fields from pIn,
//    processes the data, and returns a response in pOut.
//
//    Result:
//       FALSE - OrderStatus processed successfully.
//       TRUE - OrderStatus processing failed.
//
//================================================================
BOOL ProcessOrderStatus(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC)
{
   ORDER_STATUS_DATA * posd;
   TMON_STATE * pTMon;
   INT i;
   INT iInx;
   HRESULT hr;
   int iSize;

   pTMon = &pTPCC->tsTMon;
   posd = (ORDER_STATUS_DATA *) pTMon->pTxnData;
   ZeroMemory(posd,sizeof(ORDER_STATUS_DATA));
   posd->w_id = pTPCC->sWId;
   if (GetShortKey(&posd->d_id,pIn,"DID*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (posd->d_id < MIN_DId || posd->d_id > MAX_DId)
   {
      wsprintf(pTPCC->ErrTxt,"DId Out of Range(%ld,%ld) - %ld",
         MIN_DId,MAX_DId,posd->d_id);
      pTPCC->iStatusId = ERR_DID_INVALID;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   if (GetLongKey(&posd->c_id,pIn,"CID*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (GetStringKey(posd->c_last,pIn,"CLT*",pTPCC,NAME_LEN))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (posd->c_id == 0 && posd->c_last[0] == 0)
   {
      strcpy(pTPCC->ErrTxt,"Error - Customer Id and Name Empty");
      pTPCC->iStatusId = ERR_IDANDNAME_EMPTY;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   if (posd->c_id != 0 && posd->c_last[0] != 0)
   {
      strcpy(pTPCC->ErrTxt,
         "Error - Specify Customer Id or Name, not Both");
      pTPCC->iStatusId = ERR_IDANDNAME_ENTERED;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   iSize = pTMon->iSize;
   hr = pTMon->pIAllTxn->OrderStatus(&iSize,(unsigned char**)&pTMon-
>pTxnData);
   if (FAILED(hr))
   {
      pTPCC->iStatusId = ERR_TM_INTERFACE;
      wsprintf(pTPCC->ErrTxt,
         "COM Interface to OrderStatus Call Failed, HRESULT %x",
         hr);
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   posd = (ORDER_STATUS_DATA *) pTMon->pTxnData;
   if (posd->bTPRslt)
   {
      wsprintf(pTPCC->ErrTxt,
         "Order Status Service Returned Error(%ld): %s",
         posd->iTPRslt,posd->execution_status);
      pTPCC->iStatusId = ERR_SERVICE_RSLT;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   iInx = FormatRespHdr(pOut,"TPC-C Order-Status",pTPCC);
   iInx += wsprintf(pOut + iInx,
      "<PRE>                                  Order-Status<BR>"
      "Warehouse: %4.4d   District: %2.2d<BR>"
      "Customer: %4.4d   Name: %-16s %-2s %-16s<BR>",
      posd->w_id,posd->d_id,
      posd->c_id,posd->c_first,posd->c_middle,posd->c_last);
   iInx += sprintf(pOut + iInx,"Cust-Balance: $%9.2f<BR><BR>",posd-
>c_balance);
   iInx += wsprintf(pOut + iInx,
      "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>",
```

```
      posd->o_id,posd->o_entry_d.day,posd->o_entry_d.month,
      posd->o_entry_d.year,posd->o_entry_d.hour,
      posd->o_entry_d.minute,posd->o_entry_d.second,
      posd->o_carrier_id);
   for(i = 0; i < posd->o_ol_cnt; i++)
   {
      iInx += sprintf(pOut + iInx,
         "  %4.4d       %6.6d     %2.2d     $%8.2f      %2.2d-%2.2d-
%4.4d<BR>",
         posd->OlOrderStatusData[i].ol_supply_w_id,
         posd->OlOrderStatusData[i].ol_i_id,
         posd->OlOrderStatusData[i].ol_quantity,
         posd->OlOrderStatusData[i].ol_amount,
         posd->OlOrderStatusData[i].ol_delivery_d.day,
         posd->OlOrderStatusData[i].ol_delivery_d.month,
         posd->OlOrderStatusData[i].ol_delivery_d.year);
   };
   wsprintf(pOut + iInx,
      "<BR></PRE><HR><BR>%s</FORM>%s",szMenuList,HTMLTrailer);

   return(FALSE);

};   // ProcessOrderStatus

//=========================================================================
//
// Function name: ProcessStockLevel
//
//    ProcessStockLevel extracts the input data fields from pIn,
//    processes the data, and returns a response in pOut.
//
//    Result:
//       FALSE - StockLevel processed successfully.
//       TRUE - StockLevel processing failed.
//
//=========================================================================
BOOL ProcessStockLevel(CHAR * pIn,CHAR * pOut,TPCC_STATE * pTPCC)
{
   STOCK_LEVEL_DATA * psld;
   TMON_STATE * pTMon;
   HRESULT hr;
   INT iInx;
   int iSize;

   pTMon = &pTPCC->tsTMon;
   psld = (STOCK_LEVEL_DATA *) pTMon->pTxnData;
   ZeroMemory(psld,sizeof(STOCK_LEVEL_DATA));
   psld->w_id = pTPCC->sWId;
   psld->d_id = pTPCC->sDId;
   psld->low_stock = 0;
   psld->execution_status[0] = 0;
   if (GetShortKey(&psld->thresh_hold,pIn,"TT*",pTPCC))
   {
     FormatMenu(pOut,pTPCC);
     return(TRUE);
   };
   if (psld->thresh_hold < MIN_THRESHOLD ||
       psld->thresh_hold > MAX_THRESHOLD)
   {
      wsprintf(pTPCC->ErrTxt,"Threshold Out of Range(%ld,%ld) - %ld",
         MIN_THRESHOLD,MAX_THRESHOLD,psld->thresh_hold);
      pTPCC->iStatusId = ERR_THRESHOLD_RANGE;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   iSize = pTMon->iSize;
   hr = pTMon->pIAllTxn->StockLevel(&iSize,(unsigned char**)&pTMon-
>pTxnData);
   if (FAILED(hr))
   {
      pTPCC->iStatusId = ERR_TM_INTERFACE;
      wsprintf(pTPCC->ErrTxt,
         "COM Interface to StockLevel Call Failed, HRESULT %x",
         hr);
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   psld = (STOCK_LEVEL_DATA *) pTMon->pTxnData;
   if (psld->bTPRslt)
   {
      wsprintf(pTPCC->ErrTxt,
         "Stock Level Service Returned Error(%ld): %s",
         psld->iTPRslt,psld->execution_status);
      pTPCC->iStatusId = ERR_SERVICE_RSLT;
      FormatMenu(pOut,pTPCC);
      return(TRUE);
   };
   iInx = FormatRespHdr(pOut,"TPC-C Stock Level",pTPCC);
   wsprintf(pOut + iInx,
      "<PRE>                                Stock-Level<BR>"
      "Warehouse: %4.4d   District: %2.2d<BR><BR>"
      "Stock Level Threshold: %2.2d<BR><BR>"
      "low stock: %3.3ld</PRE><BR><HR>"
      "%s</FORM>%s",
      pTPCC->sWId,pTPCC->sDId,psld->thresh_hold,psld->low_stock,
      szMenuList,HTMLTrailer);

   return(FALSE);

};   // ProcessStockLevel

//=========================================================================
//
// Function name: GetHidden
//
//=========================================================================
BOOL GetHidden(CHAR * pMsg,UINT * uFormId,INT * iSyncId,INT * iTermId)
{
   CHAR * pPtr;
   BOOL bRslt = TRUE;

   // Extract TERMID
   pPtr = strstr(pMsg,TERMIDTOKEN);
   if (pPtr == NULL)
      goto xit;
   pPtr += strlen(TERMIDTOKEN);
   *iTermId = atoi(pPtr);

   // Extract SYNCID
   pPtr = strstr(pMsg,SYNCIDTOKEN);
   if (pPtr == NULL)
      goto xit;
```

```
   pPtr += strlen(SYNCIDTOKEN);
   *iSyncId = atoi(pPtr);

   // Extract FORMID
   pPtr = strstr(pMsg,FORMIDTOKEN);
   if (pPtr == NULL)
      goto xit;
   pPtr += strlen(FORMIDTOKEN);
   *uFormId = abs(atoi(pPtr));

   bRslt = FALSE;

xit:

   return(bRslt);

};   // GetHidden

//=========================================================================
//
// Function name: GetCmd
//
//=========================================================================
BOOL GetCmd(CHAR * pMsg,CHAR * pWork,UINT uLen)
{
   UINT u;
   CHAR * ptr;
   CHAR * pUpd;

   // Check for CMD key
   if (!(ptr = strstr(pMsg,CMDTOKEN)))
      return(CMD_NULL);
   ptr += sizeof(CMDTOKEN);
   pUpd = pWork;
   while (*ptr && *ptr != '&')
      *pUpd++ = *ptr++;
   *pUpd = 0;

   // Convert command name into command index
   for(u=0; u < CMD_MAX; u++)
   {
      if (!strcmp(szCmds[u],pWork))
         return(u);
   };

   // Command string not found
   return(CMD_NULL);

};   // GetCmd

//=========================================================================
//
// Function name: GetLongKey
//
//=========================================================================
BOOL GetLongKey(LONG * lRslt,CHAR * pHTML,CHAR * pKey,TPCC_STATE * pTPCC)
{
   if (GetKeyValue(pHTML,pKey,pTPCC->szWork,sizeof(pTPCC->szWork)))
   {
      wsprintf(pTPCC->ErrTxt,"Error - Missing %s Key",pKey);
      pTPCC->iStatusId = ERR_MISSING_KEY;
      return(TRUE);
   };
   if (pTPCC->szWork[0] != 0 )
   {
      if (CheckNumeric(pTPCC->szWork))
      {
         wsprintf(pTPCC->ErrTxt,"Error - %s Value Not Numeric",pKey);
         pTPCC->iStatusId = ERR_NOT_NUMERIC;
         return(TRUE);
      };
   };
   *lRslt = atol(pTPCC->szWork);
   return(FALSE);
};   // GetLongKey

//=========================================================================
//
// Function name: GetIntKey
//
//=========================================================================
BOOL GetIntKey(INT * iRslt,CHAR * pHTML,CHAR * pKey,TPCC_STATE * pTPCC)
{
   if (GetKeyValue(pHTML,pKey,pTPCC->szWork,sizeof(pTPCC->szWork)))
   {
      wsprintf(pTPCC->ErrTxt,"Error - Missing %s Key",pKey);
      pTPCC->iStatusId = ERR_MISSING_KEY;
      return(TRUE);
   };
   if (pTPCC->szWork[0] != 0 )
   {
      if (CheckNumeric(pTPCC->szWork))
      {
         wsprintf(pTPCC->ErrTxt,"Error - %s Value Not Numeric",pKey);
         pTPCC->iStatusId = ERR_NOT_NUMERIC;
         return(TRUE);
      };
   };
   *iRslt = atoi(pTPCC->szWork);
   return(FALSE);
};   // GetIntKey

//=========================================================================
//
// Function name: GetShortKey
//
//=========================================================================
BOOL GetShortKey(SHORT * sRslt,CHAR * pHTML,CHAR * pKey,TPCC_STATE *
pTPCC)
{
   if (GetKeyValue(pHTML,pKey,pTPCC->szWork,sizeof(pTPCC->szWork)))
   {
      wsprintf(pTPCC->ErrTxt,"Error - Missing %s Key",pKey);
      pTPCC->iStatusId = ERR_MISSING_KEY;
      return(TRUE);
   };
   if (pTPCC->szWork[0] != 0 )
   {
      if (CheckNumeric(pTPCC->szWork))
      {
         wsprintf(pTPCC->ErrTxt,"Error - %s Value Not Numeric",pKey);
         pTPCC->iStatusId = ERR_NOT_NUMERIC;
```

```
         return(TRUE);
      };
   };
   *sRslt = (SHORT) atoi(pTPCC->szWork);
   return(FALSE);
};   // GetShortKey

//========================================================================
//
// Function name: GetStringKey
//
//========================================================================
BOOL GetStringKey(CHAR * szRslt,CHAR * pHTML,CHAR * pKey,
                  TPCC_STATE * pTPCC,UINT uMax)
{
   UINT uLen;
   if (GetKeyValue(pHTML,pKey,pTPCC->szWork,sizeof(pTPCC->szWork)))
   {
      wsprintf(pTPCC->ErrTxt,"Error - Missing %s Key",pKey);
      pTPCC->iStatusId = ERR_MISSING_KEY;
      return(TRUE);
   };
   uLen = strlen(pTPCC->szWork);
   if (uLen > uMax)
   {
      wsprintf(pTPCC->ErrTxt,
         "Error - %s Key Input (%ld) Too Long (%ld)"
         ,pKey,uLen,uMax);
      pTPCC->iStatusId = ERR_INPUT_TOOLONG;
      return(TRUE);
   };
   _strupr(pTPCC->szWork);
   strcpy(szRslt,pTPCC->szWork);
   return(FALSE);
};   // GetStringKey

//========================================================================
//
// Function name: GetAmountKey
//
//========================================================================
BOOL GetAmountKey(DOUBLE * dRslt,CHAR * pHTML,CHAR * pKey,
                  TPCC_STATE * pTPCC)
{
   CHAR * ptr;
   BOOL bInvalid = FALSE;

   if (GetKeyValue(pHTML,pKey,pTPCC->szWork,sizeof(pTPCC->szWork)))
   {
      wsprintf(pTPCC->ErrTxt,"Error - Missing %s Key",pKey);
      pTPCC->iStatusId = ERR_MISSING_KEY;
      return(TRUE);
   };
   ptr = pTPCC->szWork;
   while(*ptr)
   {
      if (*ptr == '.')
      {
         ptr++;
         if (!*ptr)
            break;
         if (*ptr < '0' || *ptr > '9')
         {
            bInvalid = TRUE;
            break;
         };
         ptr++;
         if (!*ptr)
            break;
         if (*ptr < '0' || *ptr > '9')
         {
            bInvalid = TRUE;
            break;
         };
         ptr++;
         if (*ptr)
         {
            bInvalid = TRUE;
            break;
         };
         break;
      }
      else
      if (*ptr < '0' || *ptr > '9')
      {
         bInvalid = TRUE;
         break;
      };
            ptr++;
       };   // while(!*ptr)

   if (!bInvalid)
      *dRslt = atof(pTPCC->szWork);
   else
   {
      wsprintf(pTPCC->ErrTxt,
         "Error - Invalid Amount Format (%s)",pTPCC->szWork);
      pTPCC->iStatusId = ERR_AMOUNT_BADFORM;
   };

   return(bInvalid);

};   // GetAmountKey

//========================================================================
//
// Function name: GetKeyValue
//   This function parses an HTTP formatted string for specific key
//   values.  HTTP keys terminate with '='.  HTTP values terminate
//   with an '&' or '\0'.
//
//    Result:
//       FALSE - Key found, string value return in pValue
//       TRUE - Key not found
//
//========================================================================
BOOL GetKeyValue(CHAR * pHTML,CHAR * pKey,CHAR * pValue,UINT uMax)
{
   CHAR * ptr;
   if (!(ptr=strstr(pHTML,pKey)))
      return(TRUE);
   if (!(ptr=strchr(ptr,'=')))
```

```
      return(TRUE);
   ptr++;
   uMax--;
   while (*ptr && *ptr != '&' && uMax)
   {
      *pValue++ = *ptr++;
      uMax--;
   };
   *pValue = 0;
   return(FALSE);
};   // GetKeyValue

//============================================================================
//
// Function name: FormatLogin
//
//============================================================================
VOID FormatLogin(CHAR * pOut,CHAR * pAddText)
{
   wsprintf(pOut,"%s<BR>%s<BR>%s",szFormLogin,pAddText,HTMLTrailer);
};   // FormatLogin

//============================================================================
//
// Function name: FormatMenu
//
//============================================================================
VOID FormatMenu(CHAR * pOut,TPCC_STATE * pTPCC)
{
   wsprintf(pOut,
      "%s<HTML><HEAD><TITLE>TPC-C MainMenu</TITLE></HEAD><BODY>"
      "Select Desired Transaction.<BR><HR>"
      "<FORM ACTION=\"tpcc.dll\"METHOD=\"GET\">"
      "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
      "%s</FORM><BR>%s<BR>%s",
      HTTPHdr,pTPCC->iStatusId,pTPCC->iTermId,pTPCC->iSyncId,FORM_MENU,
      szMenuList,pTPCC->ErrTxt,HTMLTrailer);
};   // FormatMenu

//============================================================================
//
// Function name: FormatNewOrder
//
//============================================================================
VOID FormatNewOrder(CHAR * pOut,TPCC_STATE * pTPCC)
{
   INT iInx;
   pTPCC->uFormId = FORM_NEWORDER;
   iInx = FormatFormHdr(pOut,"TPC-C New Order",pTPCC);
   iInx += wsprintf(pOut + iInx,
      "<PRE>                                  New Order<BR>"
      "Warehouse: %4.4d   District: <INPUT NAME=\"DID*\" SIZE=1>
Date:<BR>",
      pTPCC->sWId);
       strcpy(pOut + iInx,
      "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><HR>"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Process\">"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Menu\">"
      "</FORM></BODY></HTML>");
};   // FormatNewOrder

//============================================================================
//
// Function name: FormatPayment
//
//============================================================================
VOID FormatPayment(CHAR * pOut,TPCC_STATE * pTPCC)
{
   INT iInx;
   pTPCC->uFormId = FORM_PAYMENT;
   iInx = FormatFormHdr(pOut,"TPC-C Payment",pTPCC);
   iInx += wsprintf(pOut + iInx,
      "<PRE>                                  Payment<BR>"
      "Date:<BR><BR>"
      "Warehouse: %4.4d",
      pTPCC->sWId);
   strcpy(pOut + iInx,
      "                              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></PRE><HR>"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Process\">"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Menu\">"
      "</FORM></BODY></HTML>");
};   // FormatPayment

//================================================================================
//
// Function name: FormatDelivery
//
//================================================================================
VOID FormatDelivery(CHAR * pOut,TPCC_STATE * pTPCC)
{
   INT iInx;
   pTPCC->uFormId = FORM_DELIVERY;
   iInx = FormatFormHdr(pOut,"TPC-C Delivery",pTPCC);
   wsprintf(pOut + iInx,
      "<PRE>                                  Delivery<BR>"
      "Warehouse: %4.4d<BR><BR>"
      "Carrier Number: <INPUT NAME=\"OCD*\" SIZE=1><BR><BR>"
      "Execution Status:<BR></PRE><HR>"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Process\">"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Menu\">"
      "</FORM></BODY></HTML>",
      pTPCC->sWId);
};   // FormatDelivery

//================================================================================
//
// Function name: FormatOrderStatus
//
//================================================================================
VOID FormatOrderStatus(CHAR * pOut,TPCC_STATE * pTPCC)
{
   INT iInx;
   pTPCC->uFormId = FORM_ORDERSTATUS;
   iInx = FormatFormHdr(pOut,"TPC-C Order-Status",pTPCC);
   wsprintf(pOut + iInx,
      "<PRE>                                Order-Status<BR>"
      "Warehouse: %4.4d   "
      "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></PRE><HR>"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Process\">"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Menu\">"
      "</FORM></BODY></HTML>",
      pTPCC->sWId);
};   // FormatOrderStatus

//================================================================================
//
// Function name: FormatStockLevel
//
//================================================================================
VOID FormatStockLevel(CHAR * pOut,TPCC_STATE * pTPCC)
{
   INT iInx;
   pTPCC->uFormId = FORM_STOCKLEVEL;
   iInx = FormatFormHdr(pOut,"TPC-C Stock Level",pTPCC);
   wsprintf(pOut + iInx,
      "<PRE>                                  Stock-Level<BR>"
      "Warehouse: %4.4d   District: %2.2d<BR><BR>"
      "Stock Level Threshold: <INPUT NAME=\"TT*\"SIZE=2><BR><BR>"
      "low stock:    <BR><HR>"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Process\">"
      "<INPUT TYPE=\"submit\"NAME=\"CMD\" VALUE=\"Menu\">"
      "</FORM></BODY></HTML>",
      pTPCC->sWId,pTPCC->sDId);
};   // FormatStockLevel

//================================================================================
//
// Function name: FormatFormHdr
//
//================================================================================
INT FormatFormHdr(CHAR * pOut,CHAR * pTitle,TPCC_STATE * pTPCC)
{
   return(wsprintf(pOut,
      "%s<HTML><HEAD><TITLE>%s</TITLE></HEAD>"
      "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
      "<INPUT TYPE=\"hidden\" NAME=\"PI*\" VALUE=\"\">"
      "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"
      "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">",
      HTTPHdr,pTitle,pTPCC->uFormId,pTPCC->iTermId,pTPCC->iSyncId)
      );
};   // FormatFormHdr

//================================================================================
//
// Function name: FormatRespHdr
//
//================================================================================
INT FormatRespHdr(CHAR * pOut,CHAR * pTitle,TPCC_STATE * pTPCC)
{
   return(wsprintf(pOut,
      "%s<HTML><HEAD><TITLE>%s</TITLE></HEAD>"
      "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
      "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
      "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">",
      HTTPHdr,pTitle,pTPCC->iStatusId,pTPCC->uFormId,
      pTPCC->iTermId,pTPCC->iSyncId)
      );
};   // FormatRespHdr

//================================================================================
```

```
//
// Function name: FormatString
//
//    Encodes formatted string for HTML transmission.
//
//=========================================================================
VOID FormatString(CHAR * pOut,CHAR * pPic,CHAR * pIn)
{
   while(*pPic)
   {
      if (*pPic == 'X' )
      {
         if (*pIn)
            *pOut++ = *pIn++;
         else
            *pOut++ = ' ';
      }
      else
         *pOut++ = *pPic;
      pPic++;
   };
   *pOut = 0;
};   // FormatString

//=========================================================================
// FUNCTION: UtilStrCpy
//
//    Copies n characters from string pSrc to pDst and places a null
//    null character at the end of the destination string.  Unlike
//    strncpy this function ensures that the result string is always
//    null terminated.
//
//=========================================================================
VOID UtilStrCpy(CHAR * pDest,CHAR * pSrc,INT n)
{
   strncpy(pDest,pSrc,n);
   pDest[n] = '\0';
   return;
};   // UtilStrCpy

//=========================================================================
//
// Function name: CheckNumeric
//
//    Result
//       FALSE - string is all numeric
//       TRUE - sting contains non-numeric characters
//
//=========================================================================
BOOL CheckNumeric(CHAR * pNum)
{
   if (*pNum == 0 )
      return(TRUE);
   while (*pNum && isdigit(*pNum))
      pNum++;
   return(*pNum);
};   // CheckNumeric
```

delivery.h

```
// delivery.h
//
// Copyright Unisys, 1999

#define DEFAULTDQSIZE 2000

bool DeliveryInit(long lSetThreads,long lSetQSize,char * pszPath);
void DeliveryTerm(void);
bool DeliveryPost(DELIVERY_DATA * pdd);
```

delivery.cpp

```
// delivery.cpp
//
// Copyright Unisys, 1999

#include <windows.h>
#include <stdio.h>
#include <time.h>
#include <sys\timeb.h>
#include <process.h>

#include "..\tpccsvr\tpcc.h"
#include "tmon.h"
#include "diagio.h"
#include "delivery.h"

CRITICAL_SECTION csDQRead;
CRITICAL_SECTION csDQWrite;
HANDLE hDQRead;
HANDLE hDQStart;
bool bDQStarted = FALSE;
bool bDQQuit = FALSE;
long lDeliveryThreads;
long lDQSize;
char szPath[200];
long lDeliveryActive = 0;
typedef struct
{
   bool bInUse;
   DELIVERY_DATA ddEntry;
} DELIVERY_QUEUE;
DELIVERY_QUEUE * pDQ;
long lDQNextWrite = 0;
long lDQNextRead = 0;

bool DoDQStart(void);
UINT WINAPI DoDelivery(void * Unused);
void CalculateElapsed(int * pElapsed,LPSYSTEMTIME lpBegin,
                      LPSYSTEMTIME lpEnd);

//=========================================================================
//
// Function name: DeliveryInit
//
//=========================================================================
bool DeliveryInit(long lSetThreads,long lSetQSize,char * pszPath)
```

```
{
   char szDiag[MAX_DIAG_SZ];

   lDeliveryThreads = lSetThreads;
   lDQSize = lSetQSize;
   if (lDQSize <= 0)
      lDQSize = DEFAULTDQSIZE;
   strcpy(szPath,pszPath);
   InitializeCriticalSection(&csDQRead);
   InitializeCriticalSection(&csDQWrite);
   hDQRead = CreateEvent(NULL,TRUE,FALSE,NULL);
   if (!hDQRead)
   {
      wsprintf(szDiag,"DeliveryInit: Create DQRead Event Failure (%ld)\n",
         GetLastError());
      DiagIoWrite(szDiag,DIAG_ERROR);
      return(TRUE);
   };
   pDQ = (DELIVERY_QUEUE *) calloc(lDQSize,sizeof(DELIVERY_QUEUE));
   if (pDQ == NULL)
   {
      DiagIoWrite("DeliveryInit: Allocate Delivery Queue
Failure\n",DIAG_ERROR);
      return(TRUE);
   };
   wsprintf(szDiag,
      "DeliveryInit: Threads = %ld, DQSize(entries) = %ld\n",
      lDeliveryThreads,lDQSize);
   DiagIoWrite(szDiag,DIAG_FORCE);
   return(FALSE);
};   // DeliveryInit

//============================================================================
//
// Function name: DoDQStart
//
//============================================================================
bool DoDQStart(void)
{
   UINT uThread;
   ULONG hThread;
   DWORD dwRslt;
   char szDiag[MAX_DIAG_SZ];
   void * Unused = NULL;
   int i;

   bDQStarted = TRUE;
   hDQStart = CreateEvent(NULL,TRUE,FALSE,NULL);
   if (!hDQStart)
   {
      wsprintf(szDiag,"DoDQStart: Create Event Failure (%ld)\n",
         GetLastError());
      DiagIoWrite(szDiag,DIAG_ERROR);
      return(TRUE);
   };
   for (i = 0; i < lDeliveryThreads; i++)
   {
      hThread =
         _beginthreadex(NULL,
                        0,
                        DoDelivery,
                        Unused,
                        0,
                        &uThread);
      if (hThread == 0)
      {
         wsprintf(szDiag,
            "DoDQStart: Begin Delivery Thread(%d) Failed(%ld)\n",
            i + 1,errno);
         DiagIoWrite(szDiag,DIAG_ERROR);
         return(TRUE);
      };
      dwRslt = WaitForSingleObject(hDQStart,60000);
      if (dwRslt == WAIT_TIMEOUT)
      {
         DiagIoWrite("DoDQStart: Wait Delivery Start Timed
Out\n",DIAG_ERROR);
         return(TRUE);
      };
      if (lDeliveryActive != (i + 1))
      {
         wsprintf(szDiag,
            "DoDQStart: Delivery Thread Initialization Failed(%ld)\n",
            i + 1);
         DiagIoWrite(szDiag,DIAG_ERROR);
         return(TRUE);
      };
      ResetEvent(hDQStart);
   };   // for (lDeliveryThreads)
   CloseHandle(hDQStart);
   return(FALSE);
};   // DoDQStart

//============================================================================
//
// Function name: DeliveryTerm
//
//============================================================================
void DeliveryTerm(void)
{
   int i = 0;
   bDQQuit = TRUE;
   while (i < 12 && lDeliveryActive > 0)
   {
      SetEvent(hDQRead);
      Sleep(5000);
      i++;
   };
   if (lDeliveryActive != 0)
   {
      CHAR szDiag[MAX_DIAG_SZ];
      wsprintf(szDiag,
         "DeliveryTerm: %ld DeliveryThreads still active\n",
         lDeliveryThreads);
      DiagIoWrite(szDiag,DIAG_ERROR);
   };
   free(pDQ);
   CloseHandle(hDQRead);
   DeleteCriticalSection(&csDQWrite);
   DeleteCriticalSection(&csDQRead);
   return;
};   // DeliveryTerm
```

```
//============================================================================
//
// Function name: DeliveryPost
//
//============================================================================
bool DeliveryPost(DELIVERY_DATA * pPost)
{
   DELIVERY_QUEUE * pDQSlot;
   DELIVERY_DATA * pddEntry;
   if (!bDQStarted)
   {
      if (DoDQStart())
      {
         pPost->bTPRslt = TRUE;
         pPost->iTPRslt = SVCERR_DQSTART;
         return(TRUE);
      };
   };
   __try
   {
      EnterCriticalSection(&csDQWrite);
      pDQSlot = &pDQ[lDQNextWrite];
      if (pDQSlot->bInUse)
      {
         char szDiag[MAX_DIAG_SZ];
         pPost->bTPRslt = TRUE;
         pPost->iTPRslt = SVCERR_DQFULL;
         wsprintf(szDiag,
            "Delivery Post: Queue Limit (%d) Exceeded\n",
            lDQSize);
         DiagIoWrite(szDiag,DIAG_ERROR);
         return(TRUE);
      };
      pddEntry = &pDQSlot->ddEntry;
      memcpy(pddEntry,pPost,sizeof(DELIVERY_DATA));
      pDQSlot->bInUse = TRUE;
      if (lDQNextWrite == lDQNextRead)
         SetEvent(hDQRead);
      lDQNextWrite++;
      if (lDQNextWrite == lDQSize)
         lDQNextWrite = 0;
   }
   __finally
   {
      LeaveCriticalSection(&csDQWrite);
   };
   pPost->bTPRslt = FALSE;
   pPost->iTPRslt = SVC_NOERROR;
   return(FALSE);
};   // DeliveryPost

//============================================================================
//
// Function name: DoDelivery
//
//============================================================================
UINT WINAPI DoDelivery(void * Unused)
{
   FILE *fpLog;
   char  szLogTitle[300];
   bool bFlush = FALSE;
   DELIVERY_QUEUE * pDQSlot;
   DELIVERY_DATA * pddEntry;
   DELIVERY_DATA * pdd;
   TMON_STATE tsState;
   TMON_STATE * pTMon;
   HRESULT hr;
   int iSize;
   long lMyId;
   char szTMErrTxt[500];
   char szDiag[MAX_DIAG_SZ];
   int iElapsed;
   int iInx;

   lMyId = InterlockedIncrement(&lDeliveryActive);
   pTMon = &tsState;
   pTMon->pIAllTxn = NULL;
   pTMon->pTxnData = NULL;
   pTMon->pszErrTxt = szTMErrTxt;
   if (TMInit(pTMon))
   {
      wsprintf(szDiag,"DoDelivery(%ld): TMInit %s\n",lMyId,szTMErrTxt);
      DiagIoWrite(szDiag,DIAG_ERROR);
      InterlockedDecrement(&lDeliveryActive);
      SetEvent(hDQStart);
      return(1);
   };
   wsprintf(szLogTitle,"%sdelilog%ld",szPath,lMyId);
   fpLog = fopen(szLogTitle,"ab");
       if (!fpLog)
   {
      wsprintf(szDiag,
         "DoDelivery(%ld): LogFile %s Open Failed (%ld)\n",
         lMyId,szLogTitle,GetLastError());
      DiagIoWrite(szDiag,DIAG_ERROR);
      InterlockedDecrement(&lDeliveryActive);
      SetEvent(hDQStart);
      return(1);
   };
   wsprintf(szDiag,"DoDelivery(%ld): Initialized\n",lMyId);
   DiagIoWrite(szDiag,DIAG_FORCE);
   SetEvent(hDQStart);

   while (!bDQQuit)
   {
      EnterCriticalSection(&csDQRead);
      WaitForSingleObject(hDQRead,INFINITE);
      if (bDQQuit)
      {
         LeaveCriticalSection(&csDQRead);
         break;
      };
      pDQSlot = &pDQ[lDQNextRead];
      if (!pDQSlot->bInUse)
      {
         wsprintf(szDiag,
            "DoDelivery(%ld): QSlot for Read Not InUse (%ld)\n",
         lMyId);
         DiagIoWrite(szDiag,DIAG_ERROR);
         LeaveCriticalSection(&csDQRead);
         break;
```

```
      };
      pddEntry = &pDQSlot->ddEntry;
      pdd = (DELIVERY_DATA *) pTMon->pTxnData;
      memcpy(pdd,pddEntry,sizeof(DELIVERY_DATA));
      EnterCriticalSection(&csDQWrite);
      pDQSlot->bInUse = FALSE;
      lDQNextRead++;
      if (lDQNextRead == lDQSize)
         lDQNextRead = 0;
      if (lDQNextRead == lDQNextWrite)
         ResetEvent(hDQRead);
      LeaveCriticalSection(&csDQWrite);
      LeaveCriticalSection(&csDQRead);
      // Process delivery transaction
      iSize = pTMon->iSize;
      hr = pTMon->pIAllTxn->Delivery(&iSize,(unsigned char**)&pTMon-
>pTxnData);
      if (FAILED(hr))
      {
         wsprintf(szDiag,
            "DoDelivery(%ld): COM Interface Call Failed HRESULT %x\n",
            lMyId,hr);
         DiagIoWrite(szDiag,DIAG_ERROR);
         break;
      };
      pdd = (DELIVERY_DATA *) pTMon->pTxnData;
      GetLocalTime(&pdd->EndTime);
      iElapsed = 9999999;
      if (!pdd->bTPRslt)
         CalculateElapsed(&iElapsed,&pdd->QTime,&pdd->EndTime);
      iInx =  wsprintf(szDiag,

"%2.2d/%2.2d/%2.2d,%2.2d:%2.2d:%2.2d:%3.3d,%2.2d:%2.2d:%2.2d:%3.3d,"
         "%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d\r\n",
         pdd->EndTime.wYear - 1900,pdd->EndTime.wMonth,pdd->EndTime.wDay,
         pdd->QTime.wHour,pdd->QTime.wMinute,
         pdd->QTime.wSecond,pdd->QTime.wMilliseconds,
         pdd->EndTime.wHour,pdd->EndTime.wMinute,
         pdd->EndTime.wSecond,pdd->EndTime.wMilliseconds,
         iElapsed,pdd->w_id,pdd->o_carrier_id,
         pdd->o_id[0],pdd->o_id[1],pdd->o_id[2],pdd->o_id[3],pdd->o_id[4],
         pdd->o_id[5],pdd->o_id[6],pdd->o_id[7],pdd->o_id[8],pdd->o_id[9]
);
      fwrite(szDiag,iInx,1,fpLog);
   };   // while !bDQQuit

   if (fpLog)
      fclose(fpLog);
   TMDone(pTMon);
   InterlockedDecrement(&lDeliveryActive);
   wsprintf(szDiag,"DoDelivery(%ld): Shutdown\n",lMyId);
   DiagIoWrite(szDiag,DIAG_FORCE);
   return(0);
};   // DoDelivery

//=========================================================================
//
// Function name: CalculateElapsed (milliseconds)
//
//=========================================================================
void CalculateElapsed(int * pElapsed,LPSYSTEMTIME lpBegin,
                      LPSYSTEMTIME lpEnd)
{
   int tmBegin;
   int tmEnd;
   tmBegin = (lpBegin->wHour * 3600000) + (lpBegin->wMinute * 60000) +
      (lpBegin->wSecond * 1000) + lpBegin->wMilliseconds;
   tmEnd = (lpEnd->wHour * 3600000) + (lpEnd->wMinute * 60000) +
      (lpEnd->wSecond * 1000) + lpEnd->wMilliseconds;
   *pElapsed = tmEnd - tmBegin;
   // Check for day boundry, this will function for 24 hour period but
   // will fail over a 48 hours period.
   if (*pElapsed < 0)
      *pElapsed = *pElapsed + (24 * 60 * 60 * 1000);
   return;
};   // CalculateElapsed
```

term.h

```
// term.h
//
// Copyright Unisys, 1999

#include <sys\timeb.h>

#define TMILLI_TIMEOUT 3600000       // One hour

typedef struct
{
   BOOL bInUse;                        // In use flag
   INT iTermId;                        // TermId
   LPVOID ConnID;                      // Connection Id
   INT  iSyncId;                       // Sync Id
   SHORT sWId;                         // TPCC WareHouse Id
   SHORT sDId;                         // TPCC District Id
   struct _timeb tbLastAccess;         // Last activity timestamp
} TERM_STATE;

BOOL TermInit(INT iSetMaxTerm);
VOID TermTerm(VOID);
TERM_STATE * TermAlloc(VOID);
TERM_STATE * TermGet(INT iTermId);
BOOL TermFree(INT iTermId);
```

term.cpp

```
// term.cpp
//
// Copyright Unisys, 1999
//
#include <windows.h>
#include <stdio.h>
#include "diagio.h"
#include "timesupp.h"
#include "term.h"

TERM_STATE * pTArray;
INT iNextTerm = 0;
```

```
INT iMaxTerm = 0;
CRITICAL_SECTION csTerm;

VOID TermMaint(VOID);

//============================================================================
//
// Function name: TermInit
//    Creates and initializes the first TERMINITIAL TArray entries.
//    Initializes critical section to control access to TArray.  Assumes
//    access to function is single threaded, no other threads will start
//    until this function completes and that function is called once
//    (DLL_PROCESS_ATTACH).
//
// Returns:
//    FALSE TArray allocated and initialized
//    TRUE  TArray allocation failure
//
//============================================================================
BOOL TermInit(INT iSetMaxTerm)
{
   INT iTermId;
   CHAR szDiag[MAX_DIAG_SZ];
   if (pTArray != NULL)
   {
      wsprintf(szDiag,"TermInit(%ld): TArray Already Initialized\n",
         GetCurrentThreadId());
      DiagIoWrite(szDiag,DIAG_ERROR);
      return(TRUE);
   };
   InitializeCriticalSection(&csTerm);
   iMaxTerm = iSetMaxTerm;
   pTArray = (TERM_STATE *) malloc(sizeof(TERM_STATE) * (iMaxTerm + 1));
   if (pTArray == NULL)
   {
      wsprintf(szDiag,"TermInit(%ld): malloc failed (%ld)\n",
         GetCurrentThreadId(),GetLastError());
      DiagIoWrite(szDiag,DIAG_ERROR);
      return(TRUE);
   }
   for (iTermId = 1; iTermId <= iMaxTerm; iTermId++)
      TermFree(iTermId);
   iNextTerm = 1;
   return(FALSE);
};   // TermInit

//============================================================================
//
// Function name: TermTerm
//    Frees TArray and deletes csTerm critical section.  Assumes access
//    to function is single threaded and no other threads are actively
//    accessing TArray entries (DLL_PROCESS_DETACH).
//
//============================================================================
VOID TermTerm(VOID)
{
   DeleteCriticalSection(&csTerm);
   if (pTArray != NULL)
      free(pTArray);
   iNextTerm = 0;
   iMaxTerm =0;
};   // TermTerm

//============================================================================
//
// Function name: TermAlloc
//    Allocates empty TArray.  Uses iNextTerm to start search.
//
// Returns:
//    > 0   TArray entry index (iTermId)
//    < 0   Empty TArray entry not available
//
//============================================================================
TERM_STATE * TermAlloc(VOID)
{
   INT iTermId = -1;
   if (pTArray == NULL)
   {
      CHAR szDiag[MAX_DIAG_SZ];
      wsprintf(szDiag,"TermAlloc(%ld): Term Array Not Allocated\n",
         GetCurrentThreadId());
      DiagIoWrite(szDiag,DIAG_ERROR);
      return(NULL);
   };
   EnterCriticalSection(&csTerm);
   __try
   {
      while(iNextTerm <= iMaxTerm)
      {
         if (!pTArray[iNextTerm].bInUse)
         {
            pTArray[iNextTerm].bInUse = TRUE;
            _ftime(&pTArray[iNextTerm].tbLastAccess);
            iTermId = iNextTerm;
            iNextTerm++;
            break;
         };
         iNextTerm++;
      };   // while(iNextTerm <= iMaxTerm) (1st Try)
      if (iTermId <= 0)
      {
         // No entry found.  Perform maint and try again
         TermMaint();
         iNextTerm = 1;
         while(iNextTerm <= iMaxTerm)
         {
            if (!pTArray[iNextTerm].bInUse)
            {
               pTArray[iNextTerm].bInUse = TRUE;
               _ftime(&pTArray[iNextTerm].tbLastAccess);
               iTermId = iNextTerm;
               iNextTerm++;
               break;
            };
            iNextTerm++;
         };   // while(iNextTerm <= iMaxTerm) (2nd Try)
      };   // if (iTermId <= 0)
      if (iTermId <= 0)
         iNextTerm = 1;
   }
   __finally
   {
```

```
      LeaveCriticalSection(&csTerm);
   };

   if (iTermId > 0)
      return(&pTArray[iTermId]);
   else
      return(NULL);


};   // TermAlloc

//=========================================================================
//
// Function name: TermMaint
//    Clears entries whose last access time exceeds TMILLI_TIMEOUT.
//    Assumes caller has entered csTerm.
//
//=========================================================================
VOID TermMaint(VOID)
{
   INT iTermId;
   TMILLI tmElapsed;
   // Free entries that have timed out
   for (iTermId = 1; iTermId <= iMaxTerm; iTermId++)
   {
      if (pTArray[iTermId].bInUse)
      {
         tmElapsed = TimebElapsed(&pTArray[iTermId].tbLastAccess);
         if (tmElapsed > TMILLI_TIMEOUT)
            TermFree(iTermId);
      };
   };
};   // TermMaint

//=========================================================================
//
// Function name: TermGet
//    Returns pointer to TArray slot at iTermId.
//
// Returns:
//    FALSE TArray entry made available
//    TRUE  iTermId invalid.
//
//=========================================================================
TERM_STATE * TermGet(INT iTermId)
{
   TERM_STATE * pTerm;
   TMILLI tmElapsed;
   if (iTermId <= 0 || iTermId > iMaxTerm)
   {
      CHAR szDiag[MAX_DIAG_SZ];
      wsprintf(szDiag,"TermGet(%ld): Invalid TermId (%ld)\n",
         GetCurrentThreadId(),iTermId);
      DiagIoWrite(szDiag,DIAG_ERROR);
      return(NULL);
   };
   pTerm = &pTArray[iTermId];
   if (!pTerm->bInUse)
      return(NULL);
   tmElapsed = TimebElapsed(&pTerm->tbLastAccess);
   if (tmElapsed > TMILLI_TIMEOUT)
      return(NULL);   // Entry destined to be freed by maint
   _ftime(&pTArray[iTermId].tbLastAccess);
   return(&pTArray[iTermId]);
};   // TermGet

//=========================================================================
//
// Function name: TermFree
//    Initializes contents of TArray slot at iTermId.
//
// Returns:
//    FALSE TArray entry made available
//    TRUE  iTermId invalid.
//
//=========================================================================
BOOL TermFree(INT iTermId)
{
   TERM_STATE * pTerm;
   if (iTermId <= 0 || iTermId > iMaxTerm)
   {
      CHAR szDiag[MAX_DIAG_SZ];
      wsprintf(szDiag,"TermFree(%ld): Invalid TermId (%ld)\n",
         GetCurrentThreadId(),iTermId);
      DiagIoWrite(szDiag,DIAG_ERROR);
      return(TRUE);
   };
   pTerm = &pTArray[iTermId];
   pTerm->ConnID = 0;
   pTerm->sWId = 0;
   pTerm->sDId = 0;
   pTerm->iSyncId = 0;
   pTerm->iTermId = iTermId;
   TimebClear(&pTerm->tbLastAccess);
   pTerm->bInUse = FALSE;
   return(FALSE);
};   // TermFree
```

tmon.h

```
// tmon.h
//
// Copyright Unisys, 1999

#include "..\tpccproxy\tpccproxy.h"

typedef struct
{
   CHAR * pszErrTxt;                    // Error text
   INT iSize;
   ITPCC * pIAllTxn;
   CHAR * pTxnData;                     // TM buffer area
} TMON_STATE;

VOID TMonInit(INT iSetMaxMsg);
VOID TMonTerm(VOID);
BOOL TMInit(TMON_STATE * pTMon);
VOID TMDone(TMON_STATE * pTMon);
```

tmon.cpp

```
// tmon.cpp
//
// Copyright Unisys, 1999
//

// needed for CoinitializeEx
#define _WIN32_WINNT 0x0400

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

#include "..\tpccproxy\tpccproxy_i.c"
#include "..\tpccsvr\tpccsvr_i.c"

#include "tmon.h"

INT iTMMaxSz;

//============================================================================
//
// Function name: TMonInit
//
//============================================================================
VOID TMonInit(INT iSetMaxMsg)
{
   iTMMaxSz = iSetMaxMsg;
};   // TMonInit


//============================================================================
//
// Function name: TMonTerm
//
//============================================================================
VOID TMonTerm(VOID)
{
};   // TMonTerm

//============================================================================
//
// Function name: TMInit
//
//    Result:
//       FALSE    Initialization completed successfully
//       TRUE     Initialization failed
//
//============================================================================
BOOL TMInit(TMON_STATE * pTMon)
{
   HRESULT hr = NULL;
   long lRet = 0;

   pTMon->pIAllTxn = NULL;
   pTMon->pTxnData = NULL;
   pTMon->iSize = 0;
   // Must have ErrTxt message area set before init
   if (pTMon->pszErrTxt == NULL)
      return(TRUE);

   hr = CoInitializeEx(NULL,COINIT_MULTITHREADED);
   if (FAILED(hr))
   {
      wsprintf(pTMon->pszErrTxt,"COM Initialize Failed, HRESULT %x\n",hr);
      return(TRUE);
   };

   hr = CoCreateInstance(CLSID_TPCC,NULL,CLSCTX_SERVER,IID_ITPCC,
      (void **)&pTMon->pIAllTxn);
   if (FAILED(hr))
   {
      wsprintf(pTMon->pszErrTxt,"COM Create Instance Failed, HRESULT
%x\n",hr);
      return(TRUE);
   };
   hr = (pTMon->pIAllTxn)->CallSetComplete();
   if (FAILED(hr))
   {
      wsprintf(pTMon->pszErrTxt,"COM Call SetComplete Failed, HRESULT
%x\n",hr);
      return(TRUE);
   };
   pTMon->pTxnData = (char *) CoTaskMemAlloc(iTMMaxSz);
   if (!(pTMon->pTxnData))
   {
      wsprintf(pTMon->pszErrTxt,"COM Allocate TxnData Failed\n");
      return(TRUE);
   };
   pTMon->iSize = iTMMaxSz;
   return(FALSE);
};   // TMInit

//============================================================================
//
// Function name: TMDone
//
//============================================================================
VOID TMDone(TMON_STATE * pTMon)
{
   CoTaskMemFree(pTMon->pTxnData);
   pTMon->pIAllTxn->Release();
   pTMon->pIAllTxn = NULL;
   CoUninitialize();
};   // TMDone
```

timesupp.h

```
// timesupp.h
//
// Copyright Unisys, 1999

#include <windows.h>
#include <time.h>
#include <sys\timeb.h>

#define TIMEBSEED_MOD 10000
#define TIMEBSEED_SHIFT 1000
#define TIMEB_STRING_SZ 23
```

```
#define TIMEB_STRING_DATESZ 10
#define TIMEB_STRING_TIMEOFFSET 11
#define TIMEB_STRING_TIMESZ 12

typedef ULONG TMILLI;

TMILLI TimebDiff(struct _timeb * p_tb1, struct _timeb * p_tb2);
VOID TimebCopy(struct _timeb * p_tbDest, struct _timeb * p_tbSource);
TMILLI TimebElapsed(struct _timeb * p_tb1);
VOID TimebClear(struct _timeb * p_tb1);
CHAR * TimebToString(struct _timeb * p_tb1,CHAR * psz,BOOL bMillis);
BOOL TimebFromString(struct _timeb * p_tb1,CHAR * psz);
VOID TimebAddSecs(struct _timeb * p_tb1,INT iSeconds);
ULONG TimebSeed(VOID);
```

timesupp.cpp

```
// timesupp.cpp
//
// Copyright Unisys, 1999
//

#include <stdio.h>
#include "timesupp.h"

//=========================================================================
//
// Function name: TimebCopy
//    Structure contents copy of _timeb source to _timeb dest.
//
//=========================================================================
VOID TimebCopy(struct _timeb * p_tbDest, struct _timeb * p_tbSource)
{
   p_tbDest->time = p_tbSource->time;
   p_tbDest->millitm = p_tbSource->millitm;
   p_tbDest->dstflag = p_tbSource->dstflag;
   p_tbDest->timezone = p_tbSource->timezone;

};   // TimebCopy

//=========================================================================
//
// Function name: TimebDiff
//    Time difference in milliseconds between _timeb _t1 and _timeb _t2.
//
//=========================================================================
TMILLI TimebDiff(struct _timeb * p_tb1, struct _timeb * p_tb2)
{
   LONG lRslt;
   lRslt = ((p_tb2->time - p_tb1->time) * 1000) +
               (p_tb2->millitm - p_tb1->millitm);
   if (lRslt < 0)
      return(0);
   else
      return((TMILLI) lRslt);

};   // TimebDiff

//=========================================================================
//
// Function name: TimebElaspsed
//
//=========================================================================
TMILLI TimebElapsed(struct _timeb * p_tb1)
{
   struct _timeb _tb2;
   _ftime(&_tb2);
   return (TimebDiff(p_tb1,&_tb2));

};   // TimebElapsed

//=========================================================================
//
// Function name: TimebClear
//
//=========================================================================
VOID TimebClear(struct _timeb * p_tb1)
{
   p_tb1->time = 0;
   p_tb1->millitm = 0;

};   // TimebClear

//=========================================================================
//
// Function name: TimebToString
//    Converts timeb to yyyy:mm:dd,hh:mm:ss.sss format
//
//=========================================================================
CHAR * TimebToString(struct _timeb * p_tb1,CHAR * psz,BOOL bMillis)
{
   struct tm * ptm;
   int iInx;
   ptm = localtime(&p_tb1->time);
   iInx = wsprintf(psz,"%4.4d/%2.2d/%2.2d,%2.2d:%2.2d:%2.2d",
      ptm->tm_year + 1900,ptm->tm_mon + 1,ptm->tm_mday,
      ptm->tm_hour,ptm->tm_min,ptm->tm_sec);
   if (bMillis)
      wsprintf(psz + iInx,".%3.3d",p_tb1->millitm);
   return(psz);
};   // TimebToString

//=========================================================================
//
// Function name: TimebFromString
//    Converts yyyy:mm:dd,hh:mm:ss.sss (TimebToString) format to timeb
//
//=========================================================================
BOOL TimebFromString(struct _timeb * p_tb1,CHAR * psz)
{
   struct tm tmTime;
   struct tm * ptm;
   UINT uLen;

   ptm = &tmTime;
   uLen = strlen(psz);
   if (uLen < (TIMEB_STRING_SZ - 4))   // millis are optional
   {
      p_tb1->time = 0;
      p_tb1->millitm = 0;
```

```
      return (TRUE);
   };
   // Clear fields that won't be set
   ptm->tm_wday = 0;
   ptm->tm_yday = 0;
   ptm->tm_isdst = -1;
   // Set tm struct fields from string
   ptm->tm_year = (atoi(psz)) - 1900;
   psz += 5;
   ptm->tm_mon = (atoi(psz)) - 1;
   psz += 3;
   ptm->tm_mday = atoi(psz);
   psz += 3;
   ptm->tm_hour = atoi(psz);
   psz += 3;
   ptm->tm_min = atoi(psz);
   psz +=3;
   ptm->tm_sec = atoi(psz);
   if (uLen >= TIMEB_STRING_SZ)   // Millis present
   {
      psz += 3;
      p_tb1->millitm = atoi(psz);
   };
   p_tb1->time = mktime(ptm);
   return(FALSE);
};   // TimebFromString

//=========================================================================
//
// Function name: TimebAddSecs
//
//=========================================================================
VOID TimebAddSecs(struct _timeb * p_tb1,INT iSeconds)
{
   p_tb1->time += iSeconds;
};   // TimebAddSecs

//=========================================================================
//
// Function name: TimebSeed
//
//=========================================================================
ULONG TimebSeed(VOID)
{
   ULONG ulSeed;
   struct _timeb tb_1;
   _ftime(&tb_1);
   ulSeed = ((tb_1.time % TIMEBSEED_MOD) * TIMEBSEED_SHIFT) +
tb_1.millitm;
   return(ulSeed);
};   // TimebSeed
```

diagio.h

```
// diagio.h
//
// Copyright Unisys, 1999

// Environment variable defaults
#define DEFAULTDIAGLEVEL DIAG_INFO
#define DEFAULTEVENTLOG 0

#define DIAGNOSTICS TRUE
#define MAX_DIAG_SZ 2000

// Severity level of diagnostic report
#define DIAG_FORCE 1
#define DIAG_ERROR 2
#define DIAG_STATE 3
#define DIAG_INFO 4

VOID DiagIoInit(CHAR * pDiagId,BOOL bConsole,BOOL bEvent,UINT uLevel);
VOID DiagIoTerm(VOID);
VOID DiagIoWrite(CHAR * pDiagBuffer, UINT uSeverity);
```

diagio.cpp

```
// diagio.cpp
//
// Copyright Unisys, 1999
//
#include <windows.h>
#include <stdio.h>
#include "diagio.h"

CRITICAL_SECTION csDiagIo;
HANDLE hEventLog = NULL;
UINT uDiagLevel;
BOOL bEventLog;
BOOL bConsoleLog;
CHAR * pDiagHdr;
CHAR * pEventHost;
CHAR * pErrHdr =
   {"*** ERROR *** ERROR *** ERROR *** ERROR *** ERROR ***"};

INT WriteEventLog(CHAR * pDMsgs[],UINT uMsgCnt,UINT uSeverity);

//=========================================================================
//
// Function name: DiagIoInit
//
//=========================================================================
VOID DiagIoInit(CHAR * pDiagId,BOOL bConsole,BOOL bEvent,UINT uLevel)
{
   if (DIAGNOSTICS)
   {
      InitializeCriticalSection(&csDiagIo);

      uDiagLevel = uLevel;
      bEventLog = bEvent;
      bConsoleLog = bConsole;
      pEventHost = (CHAR *) malloc(10);
      strcpy(pEventHost,"");  // local host
      pDiagHdr = (CHAR *) malloc(strlen(pDiagId) + 1);
      strcpy(pDiagHdr,pDiagId);
      if (bEventLog)
      {
         hEventLog = RegisterEventSource(pEventHost,pDiagId);
```

```
         if (hEventLog == NULL)
         {
            bEventLog = FALSE;
            if (bConsoleLog)
               fprintf(stdout,
                  "%s: Event Log Register Failed (%ld)\n"
                  "Event Log Will NOT be Used\n",
                  pDiagHdr,GetLastError());
         }
         else
         {
            if (bConsoleLog)
               fprintf(stdout,"%s: Event Logging to LocalHost as %s\n",
                  pDiagHdr,pDiagHdr);
         };
      };   // if bEventLog
   };   // if Diagnostics
};   // DiagIoInit


//============================================================================
//
// Function name: DiagIoTerm
//
//============================================================================
VOID DiagIoTerm(VOID)
{
   if (DIAGNOSTICS)
   {
      DeleteCriticalSection(&csDiagIo);
      if (hEventLog != NULL)
         DeregisterEventSource(hEventLog);
      free(pDiagHdr);
      free(pEventHost);
   };
};   // DiagIoTerm

//============================================================================
//
// Function name: DiagIoWrite
//
//============================================================================
VOID DiagIoWrite(CHAR * pDiagBuffer, UINT uSeverity)
{
   CHAR * pDMsgs[3];
   UINT uMsgCnt = 0;
   INT iERslt = 0;
   if (DIAGNOSTICS)
   {
      if (uDiagLevel >= uSeverity)
      {
         EnterCriticalSection(&csDiagIo);
         __try
         {
            if (uSeverity == DIAG_ERROR)
            {
               pDMsgs[0] = pDiagHdr;
               pDMsgs[1] = pErrHdr;
               pDMsgs[2] = pDiagBuffer;
               uMsgCnt = 3;
            }
            else
            {
               pDMsgs[0] = pDiagHdr;
               pDMsgs[1] = pDiagBuffer;
               uMsgCnt = 2;
            };
            if (bEventLog)
               iERslt = WriteEventLog(pDMsgs,uMsgCnt,uSeverity);
            if (bConsoleLog)
            {
               if (uMsgCnt == 3)
                  fprintf(stdout,"\n%s:
%s\n%s",pDMsgs[0],pDMsgs[1],pDMsgs[2]);
               else
                  fprintf(stdout,"\n%s: %s",pDMsgs[0],pDMsgs[1]);
               if (iERslt != 0)
                  fprintf(stdout,
                     "EventLog Write Failed (%ld), No Longer in Use\n",
                     iERslt);
            };
         }
         __finally
         {
            LeaveCriticalSection(&csDiagIo);
         };
      };   // if uDiagLevel >= uSeverity
   };   // if Diagnostics
};   // DiagIoWrite

INT WriteEventLog(CHAR * pDMsgs[],UINT uMsgCnt,UINT uSeverity)
{
   WORD wType;
   WORD wCount;
   wCount = uMsgCnt;
   switch (uSeverity)
   {
      case DIAG_ERROR:
         wType = EVENTLOG_ERROR_TYPE;
         break;
      default:
         wType = EVENTLOG_INFORMATION_TYPE;
         break;
   };
   if (wType != 0)
   {
      if (!ReportEvent(hEventLog,           // event log handle
                       wType,               // event type
                       0,                   // category zero
                       uSeverity,           // no event identifier
                       NULL,                // no user security identifier
                       wCount,              // # of substitution strings
                       0,                   // no binary data
                       (LPCTSTR *) pDMsgs,  // address of string array
                       NULL))               // address of binary
      {
         DeregisterEventSource(hEventLog);
         hEventLog = NULL;
         bEventLog = FALSE;
         return(GetLastError());
      };   // ReportEvent failed
   };   // if wType != 0
```

```
   return(0);
};   // WriteEventLog
```

tpccproxy.def

```
LIBRARY      "tpcc_com_ps"

DESCRIPTION  'Proxy/Stub DLL'

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

tpccproxy.idl

```
// tpccproxy.idl
//
// Copyright Unisys, 1999
// Copyright Microsoft, 1999

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

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

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

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

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

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

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

         HRESULT _stdcall CallSetComplete
            (
            );

   };   // interface ITPCC
```

tpccsvr.def

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

LIBRARY      "tpcc_com_all.dll"

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

tpccsvr.idl

```
// tpccsvr.idl
//
// Copyright Unisys, 1999
// Copyright Microsoft, 1999

interface TPCC;

import "oaidl.idl";
import "ocidl.idl";
import "..\tpccproxy\tpccproxy.idl";

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

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

```
            [default] interface ITPCC;
        };

};
```

tpccsvr.h

```
// tpccsvr.h
//
// Copyright Unisys, 1999

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

   CTPCC_Common();
   ~CTPCC_Common();

   // ITPCC
   public:
      HRESULT __stdcall NewOrder(int * iSize,UCHAR ** pTData);
      HRESULT __stdcall Payment(int * iSize,UCHAR ** pTData);
      HRESULT __stdcall Delivery(int * iSize,UCHAR ** pTData);
      HRESULT __stdcall StockLevel(   int* iSize, UCHAR ** pTData);
      HRESULT __stdcall OrderStatus(  int* iSize, UCHAR ** pTData);
      HRESULT __stdcall CallSetComplete();

   // IObjectControl
      STDMETHODIMP_(BOOL) CanBePooled() {return m_bCanBePooled;}
      STDMETHODIMP Activate() {return S_OK;} // no transactions enlistment
      STDMETHODIMP_(void) Deactivate() {   }

   // IObjectConstruct
      STDMETHODIMP Construct(IDispatch * pUnk);

       // state
   private:
      bool m_bCanBePooled;
      long m_lRefId;
      PDBPROCESS m_dbproc;
      bool m_bFailed;
      bool m_bDeadlock;
      int m_iMaxRetry;
      char m_szDBErrTxt[500];

   public:
      int HandleDbLibErr(int severity,int dberr,int oserr,
         LPCSTR dberrstr,LPCSTR oserrstr);
      int HandleSQLErr(DBINT msgno,int msgstate,int severity,
         LPCSTR msgtext);
};   // Class CTPCC_Common

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

   BEGIN_COM_MAP(CTPCC)
      COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
      COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
   END_COM_MAP()
};   // Class CTPCC
```

tpccsvr.cpp

```
// tpccsvr.cpp
//
// Copyright Unisys, 1999
// Copyright Microsoft, 1999

#define STRICT
#define _WIN32_WINNT 0x0400
#define _ATL_APARTMENT_THREADED

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

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

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

#include "tpccproxy.h"
#include "tpcc.h"

#include "resource.h"
#include "tpccsvr_i.h"
#include "tpccsvr_i.c"
#include "tpccsvr.h"
#include "..\tpccproxy\tpccproxy_i.c"

CComModule _Module;

BEGIN_OBJECT_MAP(ObjectMap)
       OBJECT_ENTRY(CLSID_TPCC, CTPCC)
END_OBJECT_MAP()
```

```
char * pProgId = "TPCC Server";
char szServer[100]     = "hostname";
char szUser[100]               = "sa";
char szMyHost[MAX_COMPUTERNAME_LENGTH + 1];
char szPassword[100]   = "";
char szDatabase[100]   = "tpcc";
int iDeadlockRetry     = 5;

long lCount = 0;
long lActive = 0;

bool ReadRegistry(VOID);
void WriteEventLog(char * pMsg,bool bError);

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

extern "C"
BOOL WINAPI DllMain(HINSTANCE hInst,ULONG ul_reason_for_call,LPVOID
lpReserved)
{
   char szDiag[300];
   DWORD dwCNSize = MAX_COMPUTERNAME_LENGTH + 1;
   try
   {
      switch(ul_reason_for_call)
      {
         case DLL_PROCESS_ATTACH:
                     _Module.Init(ObjectMap,hInst);
                     DisableThreadLibraryCalls(hInst);
            if (ReadRegistry())
            {
               WriteEventLog("DllMain: Registry Key Not Present\n",TRUE);
               return(FALSE);
            };
            GetComputerName(szMyHost,&dwCNSize);
            szMyHost[dwCNSize] = 0;
            wsprintf(szDiag,"DllMain(%s): Initialization Complete\n"
               "ServerName=%s,DB=%s,User=%s,PW=%s,Retries=%d\n",

VERSIONINFO,szServer,szDatabase,szUser,szPassword,iDeadlockRetry);
            WriteEventLog(szDiag,FALSE);
            break;

               case DLL_PROCESS_DETACH:
            WriteEventLog("DllMain: Closing down for Process
Detach\n",FALSE);
            // Signal delivery threads
            dbexit();
            _Module.Term();
            break;
      };   // switch ul_reason_for_call
   }
   catch (...)
   {
      wsprintf(szDiag,"DllMain: Unhandled exception during %s call\n",
         ul_reason_for_call == DLL_PROCESS_ATTACH ? "ATTACH" : "DETACH");
      WriteEventLog(szDiag,TRUE);
      return FALSE;
   };

   return TRUE;

};   // DllMain

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

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

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

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

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

CTPCC_Common::CTPCC_Common()
{
   char szDiag[300];
   m_bCanBePooled = TRUE;
   m_dbproc = NULL;
   m_iMaxRetry = iDeadlockRetry;
   m_lRefId = InterlockedIncrement(&lCount);
   InterlockedIncrement(&lActive);
   wsprintf(szDiag,"CTPCC_Common: Initialized %ld\n",m_lRefId);
   WriteEventLog(szDiag,FALSE);
};

CTPCC_Common::~CTPCC_Common()
{
   char szDiag[300];
   dbclose(m_dbproc);
   InterlockedDecrement(&lActive);
   wsprintf(szDiag,"~CTPCC_Common(%ld): Database closed\n",m_lRefId);
   WriteEventLog(szDiag,FALSE);
};

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

```
   HRESULT hr = CoGetObjectContext( IID_IObjectContext, (void
**)&pObjectContext );
   pObjectContext->SetComplete();
   ReleaseInterface(pObjectContext);
   return hr;
};

//========================================================================
// FUNCTION: err_handler
//
//    Handles DB-Library errors
//
// ARGUMENTS:
//    DBPROCESS   *dbproc     DBPROCESS id pointer
//    int         severity    severity of error
//    int         dberr       error id
//    int         oserr       operating system specific error code
//    char        *dberrstr   printable error description of dberr
//    char        *oserrstr   printable error description of oserr
//
// RETURNS:
//    int      INT_CANCEL
//
// COMMENTS:   None
//
//========================================================================
int err_handler(DBPROCESS *dbproc,int severity,int dberr,
               int oserr,LPCSTR dberrstr,LPCSTR oserrstr)
{
   CTPCC_Common * pCSvr;
   char szDiag[1000];
   int iRslt;
   if ((dbproc == NULL) || (DBDEAD(dbproc)))
   {
      wsprintf(szDiag,"ErrHandler: DBPROC is invalid (%s)(%d,%d,%d,%s)\n",
         dberrstr,severity,dberr,oserr,oserrstr);
      WriteEventLog(szDiag,(severity != 0));
      return(INT_CANCEL);
   };
   pCSvr = (CTPCC_Common * ) dbgetuserdata(dbproc);
   if (pCSvr == NULL)
   {
      wsprintf(szDiag,"ErrHandler: dbuserdata is invalid
(%s)(%d,%d,%d,%s)\n",
         dberrstr,severity,dberr,oserr,oserrstr);
      WriteEventLog(szDiag,(severity != 0));
      return(INT_CANCEL);
   };
   iRslt = pCSvr->HandleDbLibErr(severity,dberr,oserr,dberrstr,oserrstr);
       return(iRslt);
};   // err_handler

//========================================================================
// FUNCTION: msg_handler
//
//    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 operation
//                   INT_CANCEL      cancel operation
//
// COMMENTS:   This function also sets the dead lock dbproc
//             variable if necessary.
//
//========================================================================
int msg_handler(DBPROCESS * dbproc,DBINT msgno,int msgstate,int severity,
                LPCSTR msgtext,LPCSTR srvname,LPCSTR procname,DBUSMALLINT
line)
{
   CTPCC_Common * pCSvr;
   char szDiag[1000];
   int iRslt;
   if ((dbproc == NULL) || (DBDEAD(dbproc)))
   {
      wsprintf(szDiag,"MsgHandler: DBPROC is invalid (%s)\n",msgtext);
      WriteEventLog(szDiag,TRUE);
      return(INT_CANCEL);
   };
   pCSvr = (CTPCC_Common * ) dbgetuserdata(dbproc);
   if (pCSvr == NULL)
   {
      wsprintf(szDiag,"MsgHandler: dbuserdata is invalid
(%s)(%d,%d,%d,%s,%s,%d)\n",
         msgtext,msgno,msgstate,severity,srvname,procname,line);
      WriteEventLog(szDiag,(severity != 0));
      return(INT_CANCEL);
   };
   iRslt = pCSvr->HandleSQLErr(msgno,msgstate,severity,msgtext);
       return(iRslt);
};   // msg_handler

STDMETHODIMP CTPCC_Common::Construct(IDispatch * pUnk)
{
   char szDiag[300];
   LOGINREC    *login;
   try
       {
      if (dbgetmaxprocs() < (lActive + 5))
      {
         if (dbsetmaxprocs(lActive + 10) == FAIL)
         {
            wsprintf(szDiag,
               "Construct(%ld): Extend DBLib MaxConnections failed\n",
               m_lRefId);
            WriteEventLog(szDiag,TRUE);
            return(E_FAIL);
         };
      };
      login = dblogin();
      if (login == NULL)
      {
         wsprintf(szDiag,
            "Construct(%ld): Allocate dblogin failed\n",
            m_lRefId);
         WriteEventLog(szDiag,TRUE);
         return(E_FAIL);
```

```
};
// install error and message handlers
if (dbprocmsghandle(login,msg_handler) == NULL)
{
   wsprintf(szDiag,
      "Construct(%ld): Assign msghandler failed\n",
      m_lRefId);
   WriteEventLog(szDiag,TRUE);
   return(E_FAIL);
};
if (dbprocerrhandle(login,err_handler) == NULL)
{
   wsprintf(szDiag,
      "Construct(%ld): Assign errhandler failed\n",
      m_lRefId);
   WriteEventLog(szDiag,TRUE);
   return(E_FAIL);
};
DBSETLUSER(login,szUser);
DBSETLPWD(login,szPassword);
wsprintf(szDiag,"%s-%ld",szMyHost,m_lRefId);
DBSETLHOST(login,szDiag);
DBSETLVERSION(login,DBVER60);
if (dbsetlogintime(60) == FAIL)
{
   wsprintf(szDiag,
      "Construct(%ld): Set login time limit failed\n",
      m_lRefId);
   WriteEventLog(szDiag,TRUE);
   return(E_FAIL);
};
if (dbsettime(120) == FAIL)
{
   wsprintf(szDiag,
      "Construct(%ld): Set statement execution time limit failed\n",
      m_lRefId);
   WriteEventLog(szDiag,TRUE);
   return(E_FAIL);
};
m_dbproc = dbopen(login,szServer);
dbfreelogin(login);
if (m_dbproc == NULL)
{
   wsprintf(szDiag,
      "Construct(%ld): DbOpen Failed\n",
      m_lRefId);
   WriteEventLog(szDiag,TRUE);
   return(E_FAIL);
};
dbsetuserdata(m_dbproc,(LPVOID)this);
// Use the the right database
if (dbuse(m_dbproc,szDatabase) == FAIL)
{
   wsprintf(szDiag,
      "Construct(%ld): DbUse to %s failed\n",
      m_lRefId,szDatabase);
   WriteEventLog(szDiag,TRUE);
   return(E_FAIL);
};
dbcmd(m_dbproc,"set nocount on");
if (dbsqlexec(m_dbproc) == FAIL)
      {
         wsprintf(szDiag,
            "Construct(%ld): Set nocount on failed \n",
            m_lRefId);
         WriteEventLog(szDiag,TRUE);
         return(E_FAIL);
      };
      while (dbresults(m_dbproc) != NO_MORE_RESULTS)
      {
         while (dbnextrow(m_dbproc) != NO_MORE_ROWS)
            ;
      };
      //rollback transaction abort on
      dbcmd(m_dbproc,"set XACT_ABORT ON");
      if (dbsqlexec(m_dbproc) == FAIL)
      {
         wsprintf(szDiag,
            "Construct(%ld): Set transaction abort on failed\n",
            m_lRefId);
         WriteEventLog(szDiag,TRUE);
         return(E_FAIL);
      };
      while (dbresults(m_dbproc) != NO_MORE_RESULTS)
      {
         while (dbnextrow(m_dbproc) != NO_MORE_ROWS)
            ;
      };
        }
        catch (...)
        {
      wsprintf(szDiag,
         "Construct(%ld): Unhandled exception\n",
         m_lRefId);
      WriteEventLog(szDiag,TRUE);
               return(E_FAIL);
        }

   wsprintf(szDiag,
      "Construct(%ld): Db connection initialized\n",
      m_lRefId);
   WriteEventLog(szDiag,FALSE);
       return(S_OK);
};   // Construct

int CTPCC_Common::HandleDbLibErr(int severity,int dberr,int oserr,
                                       LPCSTR dberrstr,LPCSTR oserrstr)
{
   if (m_bFailed)
      return(INT_CANCEL);
   if (oserr != DBNOERR)
   {
      wsprintf(m_szDBErrTxt,
         "HandleDbLibErr(%ld): OSErr(%ld) - %s",
         m_lRefId,oserr,oserrstr);
      WriteEventLog(m_szDBErrTxt,TRUE);
      m_bFailed = TRUE;
   };
   return(INT_CANCEL);
};   // HandleDbLibErr

int CTPCC_Common::HandleSQLErr(DBINT msgno,int msgstate,
```

```
                                int severity,LPCSTR msgtext)
{
   if (m_bFailed)
      return(INT_CANCEL);
   if ((msgno == 5701) || (msgno == 2528) ||
       (msgno == 5703) || (msgno == 6006))
      return(INT_CONTINUE);
   // deadlock message
   if (msgno == 1205)
   {
      // set the deadlock indicator
      m_bDeadlock = TRUE;
      return(INT_CONTINUE);
   };
   if (msgno == 0)
      return(INT_CONTINUE);
   else
   {
      wsprintf(m_szDBErrTxt,
         "HandleSQLErr(%ld): MsgNo(%ld) - %s",
         m_lRefId,msgno,msgtext);
      WriteEventLog(m_szDBErrTxt,TRUE);
      m_bFailed = TRUE;
   };
   return(INT_CANCEL);
};   // HandleSQLErr

//=========================================================================
// FUNCTION: UtilStrCpy
//
//    Copies n characters from string pSrc to pDst and places a null
//    null character at the end of the destination string.  Unlike
//    strncpy this function ensures that the result string is always
//    null terminated.
//
//=========================================================================
inline static void UtilStrCpy(char * pDest, const unsigned char * pSrc,
int n)
{
   strncpy(pDest,(char *)pSrc,n);
   pDest[n] = '\0';
   return;
};   // UtilStrCpy

HRESULT CTPCC_Common::NewOrder(int * iSize,UCHAR ** pTData)
{
   NEW_ORDER_DATA * pnod;
   const BYTE * pData;
   DBDATETIME datetime;
   DBDATEREC daterec;
   int iTryit;
   int iDLCount = 0;
   int i;
   DBINT commit_flag;
   RETCODE rc;

   try
   {
      pnod = (NEW_ORDER_DATA *) *pTData;
      pnod->bTPRslt = TRUE;
      pnod->iTPRslt = SVCERR_EXCEPTION;
      m_bFailed = FALSE;
      m_bDeadlock = FALSE;
      m_szDBErrTxt[0] = 0;

      for (iTryit=0; iTryit < m_iMaxRetry; iTryit++)
      {
         if (dbrpcinit(m_dbproc,"tpcc_neworder",0) == SUCCEED)
         {
            dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1,
               (BYTE *) &pnod->w_id);
            dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1,
               (BYTE *) &pnod->d_id);
            dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1,
               (BYTE *) &pnod->c_id);
            dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1,
               (BYTE *) &pnod->o_ol_cnt);

            pnod->o_all_local = 1;
            for (i = 0; i < pnod->o_ol_cnt; i++)
            {
               if (pnod->o_all_local &&
                   pnod->Ol[i].ol_supply_w_id != pnod->w_id )
               {
                  pnod->o_all_local = 0;
                  break;
               };
            };
            dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1,
               (BYTE *) &pnod->o_all_local);

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

            if (dbrpcexec(m_dbproc) == SUCCEED)
            {
               pnod->total_amount=0;
               // Get results from order line
               for (i = 0; i<pnod->o_ol_cnt; i++)
               {
                  if (((rc = dbresults(m_dbproc)) != NO_MORE_RESULTS) &&
                      (rc != FAIL))
                  {
                     if (DBROWS(m_dbproc) && (dbnumcols(m_dbproc) == 5))
                     {
                        while (dbnextrow(m_dbproc) != NO_MORE_ROWS)
                        {
                           if(pData=dbdata(m_dbproc, 1))
                              UtilStrCpy(pnod-
>Ol[i].ol_i_name,pData,dbdatlen(m_dbproc, 1));
                           if(pData=dbdata(m_dbproc, 2))
                              pnod->Ol[i].ol_stock = (*(DBSMALLINT *)
pData);
                           if(pData=dbdata(m_dbproc, 3))
```

```
                              UtilStrCpy(pnod-
>Ol[i].ol_brand_generic,pData,dbdatlen(m_dbproc, 3));
                           if(pData=dbdata(m_dbproc, 4))

dbconvert(m_dbproc,SQLNUMERIC,pData,sizeof(DBNUMERIC),
                                 SQLFLTN,(unsigned char *) &pnod-
>Ol[i].ol_i_price,8);
                           if(pData=dbdata(m_dbproc, 5))

dbconvert(m_dbproc,SQLNUMERIC,pData,sizeof(DBNUMERIC),
                                 SQLFLTN,(unsigned char *) &pnod-
>Ol[i].ol_amount,8);
                           pnod->total_amount = pnod->total_amount + pnod-
>Ol[i].ol_amount;
                        };   // while (dbnextrow)
                     };   // if (DBROWS && dbnumcols)
                  };   // if (dbresults)
               };   // for (o_ol_cnt)
               while (((rc = dbresults(m_dbproc)) != NO_MORE_RESULTS) &&
                       (rc != FAIL))
               {
                  if (DBROWS(m_dbproc) && (dbnumcols(m_dbproc) == 8))
                  {
                     while (((rc = dbnextrow(m_dbproc)) != NO_MORE_ROWS)
&&
                             (rc != FAIL))
                     {
                        if(pData=dbdata(m_dbproc, 1))

dbconvert(m_dbproc,SQLNUMERIC,pData,sizeof(DBNUMERIC),
                           SQLFLTN,(unsigned char *) &pnod->w_tax,8);
                        if(pData=dbdata(m_dbproc, 2))

dbconvert(m_dbproc,SQLNUMERIC,pData,sizeof(DBNUMERIC),
                           SQLFLTN,(unsigned char *) &pnod->d_tax,8);
                        if(pData=dbdata(m_dbproc, 3))
                           pnod->o_id = (*(DBINT *) pData);
                        if(pData=dbdata(m_dbproc, 4))
                           UtilStrCpy(pnod-
>c_last,pData,dbdatlen(m_dbproc,4));
                        if(pData=dbdata(m_dbproc, 5))

dbconvert(m_dbproc,SQLNUMERIC,pData,sizeof(DBNUMERIC),
                              SQLFLTN,(unsigned char *) &pnod-
>c_discount,8);
                        if(pData=dbdata(m_dbproc, 6))
                           UtilStrCpy(pnod-
>c_credit,pData,dbdatlen(m_dbproc,6));
                        if(pData=dbdata(m_dbproc, 7))
                        {
                           datetime = *((DBDATETIME *) pData);
                           dbdatecrack(m_dbproc,&daterec,&datetime);
                           pnod->o_entry_d.year   = daterec.year;
                           pnod->o_entry_d.month  = daterec.month;
                           pnod->o_entry_d.day    = daterec.day;
                           pnod->o_entry_d.hour   = daterec.hour;
                           pnod->o_entry_d.minute = daterec.minute;
                           pnod->o_entry_d.second = daterec.second;
                        };
                        if(pData=dbdata(m_dbproc, 8))
                           commit_flag = (*(DBTINYINT *) pData);
                     };   // while (dbnextrow)
                  };   // if (DBROWS && dbnumcols)
               };   // while (dbresults)
            };   // if (dbrpcexec)
         };   // if (dbrpcinit)
         if (m_bDeadlock)
         {
            char szDiag[300];
            iDLCount++;
            m_bDeadlock = FALSE;
            wsprintf(szDiag,
               "NewOrder(%ld): Deadlock retry (%d)\n",
               m_lRefId,iDLCount);
            WriteEventLog(szDiag,FALSE);
            Sleep(10 * iTryit);
         }
         else
         if (m_bFailed)
         {
            strcpy(pnod->execution_status,"DBTranAbort, Check Input
Data");
            pnod->iTPRslt = SVCERR_DBLIB;
            return(S_OK);
         }
         else
         if (commit_flag == 1)
         {
            pnod->total_amount = pnod->total_amount *
               ((1 + pnod->w_tax + pnod->d_tax) * (1 - pnod->c_discount));
            strcpy(pnod->execution_status,"Transaction commited.");
            pnod->bTPRslt = FALSE;
            pnod->iTPRslt = SVC_NOERROR;
            return(S_OK);
         }
         else
         {
            strcpy(pnod->execution_status,"Item number is not valid.");
            pnod->iTPRslt = SVC_BADITEMID;
            return(S_OK);
         };
      };  // for (iTryit)
      // Reaching this code means m_iMaxRetry deadlocks occurred.
      char szDiag[300];
      wsprintf(pnod->execution_status,"Hit deadlock
max(%d).",m_iMaxRetry);
      wsprintf(szDiag,
         "NewOrder(%ld): Deadlock retry limit (%d) exceeded\n",
          m_lRefId,m_iMaxRetry);
      WriteEventLog(szDiag,TRUE);
      pnod->iTPRslt = SVCERR_DEADLOCK;
        return(S_OK);
   }
        catch (...)
        {
      char szDiag[300];
      wsprintf(szDiag,"NewOrder(%ld): Unhandled exception\n",m_lRefId);
               WriteEventLog(szDiag,TRUE);
               m_bCanBePooled = FALSE;
      pnod->bTPRslt = TRUE;
      pnod->iTPRslt = SVCERR_EXCEPTION;
               return(S_OK);
```

```
      };
};   // NewOrder

HRESULT CTPCC_Common::Payment(int * iSize,UCHAR ** pTData)
{
   PAYMENT_DATA * ppd;
   const BYTE * pData;
   DBDATETIME datetime;
   DBDATEREC daterec;
   int iTryit;
   int iDLCount = 0;
   RETCODE rc;

   try
   {
      ppd = (PAYMENT_DATA *) *pTData;
      ppd->bTPRslt = TRUE;
      ppd->iTPRslt = SVCERR_EXCEPTION;
      m_bFailed = FALSE;
      m_bDeadlock = FALSE;
      m_szDBErrTxt[0] = 0;

      for (iTryit=0; iTryit < m_iMaxRetry; iTryit++)
      {
         if (dbrpcinit(m_dbproc,"tpcc_payment",0) == SUCCEED)
         {
            dbrpcparam(m_dbproc,NULL,0,SQLINT2,-1,-1,(BYTE *) &ppd->w_id);
            dbrpcparam(m_dbproc,NULL,0,SQLINT2,-1,-1,(BYTE *) &ppd-
>c_w_id);
            dbrpcparam(m_dbproc,NULL,0,SQLFLT8,-1,-1,(BYTE *) &ppd-
>h_amount);
            dbrpcparam(m_dbproc,NULL,0,SQLINT1,-1,-1,(BYTE *) &ppd->d_id);
            dbrpcparam(m_dbproc,NULL,0,SQLINT1,-1,-1,(BYTE *) &ppd-
>c_d_id);
            dbrpcparam(m_dbproc,NULL,0,SQLINT4,-1,-1,(BYTE *) &ppd->c_id);
            if (ppd->c_id == 0)
            {
               dbrpcparam(m_dbproc,NULL,0,SQLCHAR,-1,
                  strlen(ppd->c_last),(unsigned char *)ppd->c_last);
            };
         };
         if (dbrpcexec(m_dbproc) == SUCCEED)
         {
            while (((rc = dbresults(m_dbproc)) != NO_MORE_RESULTS) && (rc
!= FAIL))
            {
               if (DBROWS(m_dbproc) && (dbnumcols(m_dbproc) == 27))
               {
                  while (((rc = dbnextrow(m_dbproc)) != NO_MORE_ROWS) &&
(rc != FAIL))
                  {
                     if(pData=dbdata(m_dbproc,1))
                        ppd->c_id = *((DBINT *) pData);
                     if(pData=dbdata(m_dbproc,2))
                        UtilStrCpy(ppd-
>c_last,pData,dbdatlen(m_dbproc,2));
                     if(pData=dbdata(m_dbproc,3))
                     {
                        datetime = *((DBDATETIME *) pData);
                        dbdatecrack(m_dbproc,&daterec,&datetime);
                        ppd->h_date.year   = daterec.year;
                        ppd->h_date.month  = daterec.month;
                        ppd->h_date.day    = daterec.day;
                        ppd->h_date.hour   = daterec.hour;
                        ppd->h_date.minute = daterec.minute;
                        ppd->h_date.second = daterec.second;
                     };
                     if(pData=dbdata(m_dbproc,4))
                        UtilStrCpy(ppd-
>w_street_1,pData,dbdatlen(m_dbproc,4));
                     if(pData=dbdata(m_dbproc,5))
                        UtilStrCpy(ppd-
>w_street_2,pData,dbdatlen(m_dbproc,5));
                     if(pData=dbdata(m_dbproc,6))
                        UtilStrCpy(ppd-
>w_city,pData,dbdatlen(m_dbproc,6));
                     if(pData=dbdata(m_dbproc,7))
                        UtilStrCpy(ppd-
>w_state,pData,dbdatlen(m_dbproc,7));
                     if(pData=dbdata(m_dbproc,8))
                        UtilStrCpy(ppd->w_zip,pData,dbdatlen(m_dbproc,8));
                     if(pData=dbdata(m_dbproc,9))
                        UtilStrCpy(ppd-
>d_street_1,pData,dbdatlen(m_dbproc,9));
                     if(pData=dbdata(m_dbproc,10))
                        UtilStrCpy(ppd-
>d_street_2,pData,dbdatlen(m_dbproc,10));
                     if(pData=dbdata(m_dbproc,11))
                        UtilStrCpy(ppd-
>d_city,pData,dbdatlen(m_dbproc,11));
                     if(pData=dbdata(m_dbproc,12))
                        UtilStrCpy(ppd-
>d_state,pData,dbdatlen(m_dbproc,12));
                     if(pData=dbdata(m_dbproc,13))
                        UtilStrCpy(ppd-
>d_zip,pData,dbdatlen(m_dbproc,13));
                     if(pData=dbdata(m_dbproc,14))
                        UtilStrCpy(ppd-
>c_first,pData,dbdatlen(m_dbproc,14));
                     if(pData=dbdata(m_dbproc,15))
                        UtilStrCpy(ppd-
>c_middle,pData,dbdatlen(m_dbproc,15));
                     if(pData=dbdata(m_dbproc,16))
                        UtilStrCpy(ppd-
>c_street_1,pData,dbdatlen(m_dbproc,16));
                     if(pData=dbdata(m_dbproc,17))
                        UtilStrCpy(ppd-
>c_street_2,pData,dbdatlen(m_dbproc,17));
                     if(pData=dbdata(m_dbproc,18))
                        UtilStrCpy(ppd-
>c_city,pData,dbdatlen(m_dbproc,18));
                     if(pData=dbdata(m_dbproc,19))
                        UtilStrCpy(ppd-
>c_state,pData,dbdatlen(m_dbproc,19));
                     if(pData=dbdata(m_dbproc,20))
                        UtilStrCpy(ppd-
>c_zip,pData,dbdatlen(m_dbproc,20));
                     if(pData=dbdata(m_dbproc,21))
                        UtilStrCpy(ppd-
>c_phone,pData,dbdatlen(m_dbproc,21));
                     if(pData=dbdata(m_dbproc,22))
                     {
```

```
                        datetime = *((DBDATETIME *) pData);
                        dbdatecrack(m_dbproc,&daterec,&datetime);
                        ppd->c_since.year   = daterec.year;
                        ppd->c_since.month  = daterec.month;
                        ppd->c_since.day    = daterec.day;
                        ppd->c_since.hour   = daterec.hour;
                        ppd->c_since.minute = daterec.minute;
                        ppd->c_since.second = daterec.second;
                     };
                     if(pData=dbdata(m_dbproc,23))
                        UtilStrCpy(ppd-
>c_credit,pData,dbdatlen(m_dbproc,23));
                     if(pData=dbdata(m_dbproc,24))

dbconvert(m_dbproc,SQLNUMERIC,pData,sizeof(DBNUMERIC),
                           SQLFLTN,(unsigned char *) &ppd-
>c_credit_lim,8);
                     if(pData=dbdata(m_dbproc,25))

dbconvert(m_dbproc,SQLNUMERIC,pData,sizeof(DBNUMERIC),
                           SQLFLTN,(unsigned char *) &ppd->c_discount,8);
                     if(pData=dbdata(m_dbproc,26))

dbconvert(m_dbproc,SQLNUMERIC,pData,sizeof(DBNUMERIC),
                        SQLFLTN,(unsigned char *) &ppd->c_balance,8);
                     if(pData=dbdata(m_dbproc,27))
                        UtilStrCpy(ppd-
>c_data,pData,dbdatlen(m_dbproc,27));
                  };   // while (dbnextrow)
               };   // if (DBROWS && dbnumcols)
            };   // while (dbresults)
         };   // if (dbrpcexe)
         if (m_bDeadlock)
         {
            char szDiag[300];
            iDLCount++;
            m_bDeadlock = FALSE;
            wsprintf(szDiag,
               "Payment(%ld): Deadlock retry (%d)\n",
               m_lRefId,iDLCount);
            WriteEventLog(szDiag,FALSE);
            Sleep(10 * iTryit);
         }
         else
         if (m_bFailed)
         {
            strcpy(ppd->execution_status,"DBTranAbort, Check Input Data");
            ppd->iTPRslt = SVCERR_DBLIB;
            return(S_OK);
         }
         else
         if (ppd->c_id == 0)
         {
            strcpy(ppd->execution_status,"Invalid Customer id,name.");
            ppd->iTPRslt = SVCERR_NOCUSTOMER;
            return(S_OK);
         }
         else
         {
            strcpy(ppd->execution_status,"Transaction commited.");
            ppd->bTPRslt = FALSE;
            ppd->iTPRslt = SVC_NOERROR;
            return(S_OK);
         };
      };  // for (iTryit)
      // Reaching this code means m_iMaxRetry deadlocks occurred.
      char szDiag[300];
      wsprintf(ppd->execution_status,"Hit deadlock max(%d).",m_iMaxRetry);
      wsprintf(szDiag,
         "Payment(%ld): Deadlock retry limit (%d) exceeded\n",
         m_lRefId,m_iMaxRetry);
      WriteEventLog(szDiag,TRUE);
      ppd->iTPRslt = SVCERR_DEADLOCK;
       return(S_OK);
   }
       catch (...)
       {
      char szDiag[300];
      wsprintf(szDiag,"Payment(%ld): Unhandled exception\n",m_lRefId);
               WriteEventLog(szDiag,TRUE);
               m_bCanBePooled = FALSE;
      ppd->bTPRslt = TRUE;
      ppd->iTPRslt = SVCERR_EXCEPTION;
               return(S_OK);
       };
};   // Payment


HRESULT CTPCC_Common::Delivery(int * iSize,UCHAR ** pTData)
{
   DELIVERY_DATA * pdd;
   const BYTE * pData;
   int i;
   int iTryit;
   int iDLCount = 0;
   RETCODE rc;

   try
   {
      pdd = (DELIVERY_DATA *) *pTData;
      pdd->bTPRslt = TRUE;
      pdd->iTPRslt = SVCERR_EXCEPTION;
      m_bFailed = FALSE;
      m_bDeadlock = FALSE;
      m_szDBErrTxt[0] = 0;

      for (iTryit=0; iTryit < m_iMaxRetry; iTryit++)
      {
         if (dbrpcinit(m_dbproc,"tpcc_delivery",0) == SUCCEED)
         {
            dbrpcparam(m_dbproc,NULL,0,SQLINT2,-1,-1,(BYTE *) &pdd->w_id);
            dbrpcparam(m_dbproc,NULL,0,SQLINT1,-1,-1,(BYTE *) &pdd-
>o_carrier_id);

            if (dbrpcexec(m_dbproc) == SUCCEED)
            {
               while (((rc = dbresults(m_dbproc)) != NO_MORE_RESULTS) &&
(rc != FAIL))
               {
                  while (((rc = dbnextrow(m_dbproc)) != NO_MORE_ROWS) &&
(rc != FAIL))
                  {
```

```
                    for (i = 0; i < 10; i++)
                    {
                       if(pData = dbdata(m_dbproc,i + 1))
                          pdd->o_id[i] = *((DBINT *)pData);
                       else
                          pdd->o_id[i] = 0;
                    };
                 };   // while (dbnextrow)
              };   // while (dbresults)
           };   // if (dbrpcexec)
        };   // if (dbrpcinit)
        if (m_bDeadlock)
        {
           char szDiag[300];
           iDLCount++;
           m_bDeadlock = FALSE;
           wsprintf(szDiag,
              "Delivery(%ld): Deadlock retry (%d)\n",
              m_lRefId,iDLCount);
           WriteEventLog(szDiag,FALSE);
           Sleep(10 * iTryit);
        }
        else
        if (m_bFailed)
        {
           pdd->iTPRslt = SVCERR_DBLIB;
           return(S_OK);
        }
        else
        {
           pdd->bTPRslt = FALSE;
           pdd->iTPRslt = SVC_NOERROR;
           return(S_OK);
        };
     };  // for (iTryit)
     // Reaching this code means m_iMaxRetry deadlocks occurred.
     char szDiag[300];
     wsprintf(szDiag,
        "Payment(%ld): Deadlock retry limit (%d) exceeded\n",
        m_lRefId,m_iMaxRetry);
     WriteEventLog(szDiag,TRUE);
     pdd->iTPRslt = SVCERR_DEADLOCK;
      return(S_OK);
   }
      catch (...)
      {
     char szDiag[300];
     wsprintf(szDiag,"Delivery(%ld): Unhandled exception\n",m_lRefId);
             WriteEventLog(szDiag,TRUE);
             m_bCanBePooled = FALSE;
     pdd->bTPRslt = TRUE;
     pdd->iTPRslt = SVCERR_EXCEPTION;
             return(S_OK);
      };
};   // Delivery

HRESULT CTPCC_Common::OrderStatus(int * iSize,UCHAR ** pTData)
{
   ORDER_STATUS_DATA * posd;
   const BYTE * pData;
   DBDATETIME datetime;
   DBDATEREC daterec;
   int iTryit;
   int iDLCount = 0;
   int i;
   RETCODE rc;

   try
   {
      posd = (ORDER_STATUS_DATA *) *pTData;
      posd->bTPRslt = TRUE;
      posd->iTPRslt = SVCERR_EXCEPTION;
      m_bFailed = FALSE;
      m_bDeadlock = FALSE;
      m_szDBErrTxt[0] = 0;

      for (iTryit=0; iTryit < m_iMaxRetry; iTryit++)
      {
         if (dbrpcinit(m_dbproc,"tpcc_orderstatus", 0) == SUCCEED)
         {
            dbrpcparam(m_dbproc,NULL,0,SQLINT2,-1,-1,(BYTE *) &posd-
>w_id);
            dbrpcparam(m_dbproc,NULL,0,SQLINT1,-1,-1,(BYTE *) &posd-
>d_id);
            dbrpcparam(m_dbproc,NULL,0,SQLINT4,-1,-1,(BYTE *) &posd-
>c_id);
            if (posd->c_id == 0)
            {
               dbrpcparam(m_dbproc,NULL,0,SQLCHAR,-1,
                  strlen(posd->c_last),(unsigned char *)posd->c_last);
            };
         };
         if (dbrpcexec(m_dbproc) == SUCCEED)
         {
            while (((rc = dbresults(m_dbproc)) != NO_MORE_RESULTS) && (rc
!= FAIL))
            {
               if (DBROWS(m_dbproc) && (dbnumcols(m_dbproc) == 5))
               {
                  i = 0;
                  while (((rc = dbnextrow(m_dbproc)) != NO_MORE_ROWS) &&
(rc != FAIL))
                  {
                     if(pData=dbdata(m_dbproc,1))
                        posd->OlOrderStatusData[i].ol_supply_w_id =
(*(DBSMALLINT *) pData);
                     if(pData=dbdata(m_dbproc,2))
                        posd->OlOrderStatusData[i].ol_i_id = (*(DBINT *)
pData);
                     if(pData=dbdata(m_dbproc,3))
                        posd->OlOrderStatusData[i].ol_quantity =
(*(DBSMALLINT *) pData);
                     if(pData=dbdata(m_dbproc,4))

dbconvert(m_dbproc,SQLNUMERIC,pData,sizeof(DBNUMERIC),SQLFLTN,
                           (unsigned char *)&posd-
>OlOrderStatusData[i].ol_amount,8);
                     if(pData=dbdata(m_dbproc,5))
                     {
                        datetime = *((DBDATETIME *) pData);
                        dbdatecrack(m_dbproc,&daterec,&datetime);
```

```
                        posd->OlOrderStatusData[i].ol_delivery_d.year   =
daterec.year;
                        posd->OlOrderStatusData[i].ol_delivery_d.month  =
daterec.month;
                        posd->OlOrderStatusData[i].ol_delivery_d.day    =
daterec.day;
                        posd->OlOrderStatusData[i].ol_delivery_d.hour   =
daterec.hour;
                        posd->OlOrderStatusData[i].ol_delivery_d.minute =
daterec.minute;
                        posd->OlOrderStatusData[i].ol_delivery_d.second =
daterec.second;
                     };
                     i++;
                  };   // while (dbnextrow)
                  posd->o_ol_cnt = i;
               }   // if (DBROWS && dbnumcols == 5)
               else
               if (DBROWS(m_dbproc) && (dbnumcols(m_dbproc) == 8))
               {
                  while (((rc = dbnextrow(m_dbproc)) != NO_MORE_ROWS) &&
(rc != FAIL))
                  {
                     if(pData=dbdata(m_dbproc,1))
                        posd->c_id = (*(DBINT *) pData);
                     if(pData=dbdata(m_dbproc,2))
                        UtilStrCpy(posd-
>c_last,pData,dbdatlen(m_dbproc,2));
                     if(pData=dbdata(m_dbproc,3))
                        UtilStrCpy(posd-
>c_first,pData,dbdatlen(m_dbproc,3));
                     if(pData=dbdata(m_dbproc,4))
                        UtilStrCpy(posd-
>c_middle,pData,dbdatlen(m_dbproc,4));
                     if(pData=dbdata(m_dbproc,5))
                     {
                        datetime = *((DBDATETIME *) pData);
                        dbdatecrack(m_dbproc,&daterec,&datetime);
                        posd->o_entry_d.year   = daterec.year;
                        posd->o_entry_d.month  = daterec.month;
                        posd->o_entry_d.day    = daterec.day;
                        posd->o_entry_d.hour   = daterec.hour;
                        posd->o_entry_d.minute = daterec.minute;
                        posd->o_entry_d.second = daterec.second;
                     };
                     if(pData=dbdata(m_dbproc,6))
                        posd->o_carrier_id = (*(DBSMALLINT *) pData);
                     if(pData=dbdata(m_dbproc,7))

dbconvert(m_dbproc,SQLNUMERIC,pData,sizeof(DBNUMERIC),SQLFLTN,
                           (unsigned char *) &posd->c_balance,8);
                     if(pData=dbdata(m_dbproc,8))
                        posd->o_id = (*(DBINT *) pData);
                  };   // while (dbnextrow)
               };   // if (DBROWS && dbnumcols == 8)
               if (i==0)
               {
                  posd->iTPRslt = SVCERR_NOORDERS;
                  return(S_OK);
               };
            };   // while (dbresults)
         };   // if (dbrpcexec)
         if (m_bDeadlock)
         {
            char szDiag[300];
            iDLCount++;
            m_bDeadlock = FALSE;
            wsprintf(szDiag,
               "OrderStatus(%ld): Deadlock retry (%d)\n",
               m_lRefId,iDLCount);
            WriteEventLog(szDiag,FALSE);
            Sleep(10 * iTryit);
         }
         else
         if (m_bFailed)
         {
            strcpy(posd->execution_status,"DBTranAbort, Check Input
Data");
            posd->iTPRslt = SVCERR_DBLIB;
            return(S_OK);
         }
         else
         if (posd->c_id == 0)
         {
            strcpy(posd->execution_status,"Invalid Customer id,name.");
            posd->iTPRslt = SVCERR_NOCUSTOMER;
            return(S_OK);
         }
         else
         {
            strcpy(posd->execution_status,"Transaction commited.");
            posd->bTPRslt = FALSE;
            posd->iTPRslt = SVC_NOERROR;
            return(S_OK);
         };
      };  // for (iTryit)
      // Reaching this code means m_iMaxRetry deadlocks occurred.
      char szDiag[300];
      wsprintf(posd->execution_status,"Hit deadlock
max(%d).",m_iMaxRetry);
      wsprintf(szDiag,
         "OrderStatus(%ld): Deadlock retry limit (%d) exceeded\n",
         m_lRefId,m_iMaxRetry);
      WriteEventLog(szDiag,TRUE);
      posd->iTPRslt = SVCERR_DEADLOCK;
        return(S_OK);
   }
        catch (...)
        {
      char szDiag[300];
      wsprintf(szDiag,"OrderStatus(%ld): Unhandled exception\n",m_lRefId);
               WriteEventLog(szDiag,TRUE);
               m_bCanBePooled = FALSE;
      posd->bTPRslt = TRUE;
      posd->iTPRslt = SVCERR_EXCEPTION;
               return(S_OK);
        };
};   // Orderstatus

HRESULT CTPCC_Common::StockLevel(int * iSize,UCHAR ** pTData)
{
   STOCK_LEVEL_DATA * psld;
```

```
   const BYTE * pData;
   int iTryit;
   int iDLCount = 0;
   RETCODE rc;

   try
   {
      psld = (STOCK_LEVEL_DATA *) *pTData;
      psld->bTPRslt = TRUE;
      psld->iTPRslt = SVCERR_EXCEPTION;
      m_bFailed = FALSE;
      m_bDeadlock = FALSE;

      for (iTryit=0; iTryit < m_iMaxRetry; iTryit++)
      {
         if (dbrpcinit(m_dbproc,"tpcc_stocklevel",0) == SUCCEED)
         {
            dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1,
               (BYTE *) &psld->w_id);
            dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1,
               (BYTE *) &psld->d_id);
            dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1,
               (BYTE *) &psld->thresh_hold);

            if (dbrpcexec(m_dbproc) == SUCCEED)
            {
               while (((rc = dbresults(m_dbproc)) != NO_MORE_RESULTS) &&
                     (rc != FAIL))
               {
                  if (DBROWS(m_dbproc))
                  {
                     while (((rc = dbnextrow(m_dbproc)) != NO_MORE_ROWS)
&&
                           (rc != FAIL))
                     {
                        if(pData=dbdata(m_dbproc,1))
                           psld->low_stock = *((long *) pData);
                     };
                  };   // if (DBROWS(m_dbproc)
               };   // while (dbresults)
            };   // if (dbrpcexec)
         };   // if (dbrpcinit)
         if (m_bDeadlock)
         {
            char szDiag[300];
            iDLCount++;
            m_bDeadlock = FALSE;
            wsprintf(szDiag,
               "StockLevel(%ld): Deadlock retry (%d)\n",
               m_lRefId,iDLCount);
            WriteEventLog(szDiag,FALSE);
            Sleep(10 * iTryit);
         }
         else
         if (m_bFailed)
         {
            strcpy(psld->execution_status,"DBTranAbort, Check Input
Data");
            psld->iTPRslt = SVCERR_DBLIB;
            return(S_OK);
         }
         else
         {
            strcpy(psld->execution_status,"Transaction commited.");
            psld->bTPRslt = FALSE;
            psld->iTPRslt = SVC_NOERROR;
            return(S_OK);
         };
      };  // for (iTryit)
      // Reaching this code means m_iMaxRetry deadlocks occurred.
      char szDiag[300];
      wsprintf(psld->execution_status,"Hit deadlock
max(%d).",m_iMaxRetry);
      wsprintf(szDiag,
         "StockLevel(%ld): Deadlock retry limit (%d) exceeded\n",
         m_lRefId,m_iMaxRetry);
      WriteEventLog(szDiag,TRUE);
      psld->bTPRslt = TRUE;
      psld->iTPRslt = SVCERR_DEADLOCK;
       return(S_OK);
   }
       catch (...)
       {
      char szDiag[300];
      wsprintf(szDiag,"StockLevel(%ld): Unhandled exception\n",m_lRefId);
               WriteEventLog(szDiag,TRUE);
               m_bCanBePooled = FALSE;
      psld->bTPRslt = TRUE;
      psld->iTPRslt = SVCERR_EXCEPTION;
               return(S_OK);
       };
};   // Stocklevel

//============================================================================
//
// Function name: ReadRegistry
//
//    Sets global operational parameters from registry if they exist.
//    Otherwise, compiled in defaults apply.
//
// Result:
//    FALSE    Registry entry found
//    TRUE     Registry entry does not exist
//
//============================================================================
bool ReadRegistry(VOID)
{
   HKEY hkTPCC;
   DWORD       dwMax;
   DWORD       dwRT;
   char szValue[100];
   if (RegOpenKeyEx(HKEY_LOCAL_MACHINE,"SOFTWARE\\Unisys\\TPCC",0,
      KEY_READ, &hkTPCC) != ERROR_SUCCESS )
      return(TRUE);
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"SERVERNAME",0,&dwRT,(BYTE *)
&szValue,&dwMax)
      == ERROR_SUCCESS)
      strcpy(szServer,szValue);
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"DATABASE",0,&dwRT,(BYTE *) &szValue,&dwMax)
      == ERROR_SUCCESS)
```

```
      strcpy(szDatabase,szValue);
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"USER",0,&dwRT,(BYTE *) &szValue,&dwMax)
      == ERROR_SUCCESS)
      strcpy(szUser,szValue);
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"PASSWORD",0,&dwRT,(BYTE *) &szValue,&dwMax)
      == ERROR_SUCCESS)
      strcpy(szPassword,szValue);
   dwMax = sizeof(szValue);
   if (RegQueryValueEx(hkTPCC,"MAXRETRY",0,&dwRT,(BYTE *) &szValue,&dwMax)
      == ERROR_SUCCESS )
      iDeadlockRetry = abs(atoi(szValue));
   RegCloseKey(hkTPCC);
   return(FALSE);
};   // ReadRegistry

//=========================================================================
//
// Function name: WriteEventLog
//
//=========================================================================
void WriteEventLog(char * pMsg,bool bError)
{
   WORD wType;
   char szHeader[100];
   char * pDMsgs[2];
   HANDLE hEventLog = NULL;
   if (bError)
      wType = EVENTLOG_ERROR_TYPE;
   else
      wType = EVENTLOG_INFORMATION_TYPE;
   hEventLog = RegisterEventSource(NULL,pProgId);
   wsprintf(szHeader,"%s (%ld)\n",pProgId,GetCurrentThreadId());
   pDMsgs[0] = szHeader;
   pDMsgs[1] = pMsg;
   if (hEventLog != NULL)
   {
      ReportEvent(hEventLog,           // event log handle
                  wType,               // event type
                  0,                   // category zero
                  0,                   // no event identifier
                  NULL,                // no user security identifier
                  2,                   // # of substitution strings
                  0,                   // no binary data
                  (LPCTSTR *) pDMsgs,  // address of string array
                  NULL);               // address of binary data
      DeregisterEventSource(hEventLog);
   };
   return;
};   // WriteEventLog
```

Appendix B - Database Design

Build Scripts

BACKUP.SQL

```
--  File:     BACKUP.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Creates backup of tpcc database


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

dump database tpcc to 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
```

BACKUPDEVB.SQL

```
--  File:     BACKUPDEVB.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Creates tpcc database Backup Devices


use master
go

--  create backup devices

exec sp_addumpdevice 'disk','tpccback1','U:\tpccback1.dmp'
exec sp_addumpdevice 'disk','tpccback2','V:\tpccback2.dmp'
exec sp_addumpdevice 'disk','tpccback3','U:\tpccback3.dmp'
exec sp_addumpdevice 'disk','tpccback4','V:\tpccback4.dmp'
go
```

CREATEDB.SQL

```
--  File:     CREATEDB.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Creates tpcc database and backup files
--           for 2160 warehouse database.

use master
go

--      Create temporary table for timing

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

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

insert  into tpcc_timer values (0,0)
go

--      Store starting time

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

--  create main database files

CREATE DATABASE tpcc
ON PRIMARY
(       NAME            = MSSQL70_tpcc_root,
        FILENAME        = "C:\MSSQL7\Data\tpcc_root.mdf",
        SIZE            = 10MB,
        FILEGROWTH      = 0),
FILEGROUP       MSSQL70_misc_fg
(       NAME            = MSSQL70_misc1,
        FILENAME        = "K:",
        SIZE            = 9500MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_misc2,
        FILENAME        = "E:",
        SIZE            = 9500MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_misc3,
        FILENAME        = "P:",
        SIZE            = 9500MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_misc4,
        FILENAME        = "G:",
        SIZE            = 9500MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_misc5,
        FILENAME        = "N:",
        SIZE            = 9500MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_misc6,
        FILENAME        = "I:",
        SIZE            = 9500MB,
        FILEGROWTH      = 0),
```

```
(       NAME            = MSSQL70_misc7,
        FILENAME        = "R:",
        SIZE            = 9500MB,
        FILEGROWTH      = 0),
FILEGROUP       MSSQL70_cs_fg
(       NAME            = MSSQL70_cs1,
        FILENAME        = "Q:",
        SIZE            = 18000MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs2,
        FILENAME        = "H:",
        SIZE            = 18000MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs3,
        FILENAME        = "M:",
        SIZE            = 18000MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs4,
        FILENAME        = "F:",
        SIZE            = 18000MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs5,
        FILENAME        = "O:",
        SIZE            = 18000MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs6,
        FILENAME        = "D:",
        SIZE            = 18000MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs7,
        FILENAME        = "J:",
        SIZE            = 18000MB,
        FILEGROWTH      = 0)
LOG ON
(       NAME            =MSSQL70_tpcc_log,
        FILENAME        ="L:",
        SIZE            =68801MB,
        FILEGROWTH      =0)
go

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

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

--      remove temporary table

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

DBOPT1.SQL

```
--  File:     DBOPT1.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  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. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Resets database options after data load


use master
go

sp_dboption tpcc,'select ',false
go

sp_dboption tpcc,'trunc. ',false
go

use tpcc
go

checkpoint
go

sp_configure allow,1
go

reconfigure with override
go

/*                                              */
/* Set option values for user-defined indexes */
/*                                              */

sp_indexoption 'customer',    'AllowPageLocks',      FALSE
go
sp_indexoption 'district',    'AllowPageLocks',      FALSE
go
sp_indexoption 'warehouse',   'AllowPageLocks',      FALSE
go
sp_indexoption 'stock',       'AllowPageLocks',      FALSE
go
sp_indexoption 'order_line',  'AllowRowLocks',       FALSE
go
sp_indexoption 'orders',      'AllowRowLocks',       FALSE
```

```
go
sp_indexoption 'new_order',   'AllowRowLocks',       FALSE
go
sp_indexoption 'item',        'AllowRowLocks',       FALSE
go
sp_indexoption 'item',        'AllowPageLocks',      FALSE
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,0
go

reconfigure with override
go

exec sp_dboption tpcc, 'auto update statistics',     FALSE
exec sp_dboption tpcc, 'auto create statistics',     FALSE
go

exec sp_tableoption    "district",    "pintable",true
exec sp_tableoption    "warehouse",   "pintable",true
exec sp_tableoption    "new_order",   "pintable",true
exec sp_tableoption    "item",        "pintable",true
go
```

REMOVEDB.SQL

```
--  File:     REMOVEDB.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Removes tpcc database and backup files


use master
go

--  remove any existing database and backup files

exec sp_dbremove tpcc, dropdev
go

exec sp_dropdevice 'tpccback1'
exec sp_dropdevice 'tpccback2'
exec sp_dropdevice 'tpccback3'
exec sp_dropdevice 'tpccback4'
go
```

RESTORE.SQL

```
--  File:     RESTORE.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Loads database backup from backup files


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

load database tpcc from 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
```

RUNCFG70.SQL

```
/*  TPC-C Benchmark Kit
*/
/*
*/
/*  RUNCFG70.SQL
*/
/*
*/
/*  This script file is used to set runtime server configuration
parameters   */
/*
*/


exec sp_configure "show advanced option", 1
go

reconfigure with override
go

/* ensures sufficient I/O bandwidth is generated by SQL Server */
exec sp_configure "max async IO",255

/* change this value to approximately the number of connected users */
exec sp_configure "max worker threads",438
```

```
/* increase priority of user threads */
exec sp_configure "priority boost",1

/* disable automatic checkpointing */
exec sp_configure "recovery interval",32767

/* change to a mask appropriate for the number of processors on the server
*/
exec sp_configure "affinity mask",0xff

/* enable fibers */
exec sp_configure "lightweight pooling",1

go

reconfigure with override
go
```

SQLSHUTDOWN.SQL

```
use tpcc
go
checkpoint
go
shutdown
go
```

VERIFY_BUILD.SQL

```
--  File:     VERIFY_BUILD.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Verifies the SQL Server Version

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

create proc ms_verify_build

as

declare @version_num   char(8),
       @good_build    int

-- store the version of SQL Server you are running
select  @version_num   = SUBSTRING((select @@version),30,8)

if (select CAST(SUBSTRING(@version_num,3,2) AS int)) < 10
       begin
              if (select CAST(SUBSTRING(@version_num,6,3) AS int)) < 623
                     RAISERROR (50001,11,1)
       end
else
       begin
              if (select CAST(SUBSTRING(@version_num,6,3) AS int)) < 100
                     RAISERROR (50002,11,1)
       end
go
```

VERIFY_MSG.SQL

```
exec sp_dropmessage 50001
exec sp_dropmessage 50002
exec sp_dropmessage 50003
exec sp_addmessage 50001, 1,"Incorrect SQL Server Build - You must run
7.00.623 or higher"
exec sp_addmessage 50002, 1,"Incorrect SQL Server Build - You must run
7.10.100 or higher"
exec sp_addmessage 50003, 1,"Incorrect Sort Order - Please re-install SQL
Server with the Binary Sort Order"
```

VERIFY_SORT.SQL

```
--  File:     VERIFY_SORT.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Verifies the Sort Order

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

create proc ms_verify_sort

as

declare @sort_order    int


-- get the sort order
select  @sort_order    = (select value from sysconfigures where config =
'1123')

if (select @sort_order) <> 50
       RAISERROR (50003,11,1)
go
```

VERIFYTPCCLOAD

```
--  File:     VERIFYTPCCLOAD.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Performs series of TPCC database checks to verify
--            that database load completed correctly

print   " "
select  convert(char(30), getdate(),9)
print   " "

use tpcc
go

--      **********************************
--
```

```
--      Check rows per table from SYSINDEXES
--
--      ***********************************

print   'WAREHOUSE TABLE'

select  rows
from    sysindexes
where   id      = object_id("warehouse")
go

print   'DISTRICT TABLE = (10 * No of warehouses)'

select  rows
from    sysindexes
where   id      =object_id("district")
go

print   'ITEM TABLE = 100,000'

select  rows
from    sysindexes
where   id      =object_id("item")
go

print   'CUSTOMER TABLE = (30,000 * No of warehouses)'

select  rows
from    sysindexes
where   id      =object_id("customer")
go

print 'ORDERS TABLE = (30,000 * No of warehouses)'

select  rows
from    sysindexes
where   id      =object_id("orders")
go

print   'HISTORY TABLE = (30,000 * No of warehouses)'

select  rows
from    sysindexes
where   id      =object_id("history")
go

print   'STOCK TABLE = (100,000 * No of warehouses)'

select  rows
from    sysindexes
where   id      =object_id("stock")
go

print   'ORDER_LINE TABLE = (300,000 * No of warehouses + some change)'

select  rows
from    sysindexes
where   id      =object_id("order_line")
go

print   'NEW_ORDER TABLE = (9000 * No of warehouses)'

select  rows
from    sysindexes
where   id      =object_id("new_order")
go

--      *************
--
--      Check indices
--
--      *************

print '**************Index Check****************'

use tpcc
go

sp_helpindex   customer
go

sp_helpindex   stock
go

sp_helpindex   district
go

sp_helpindex   item
go

sp_helpindex   new_order
go

sp_helpindex   orders
go

sp_helpindex   order_line
go

sp_helpindex   warehouse
go
```

Tables

IDXCUSCL.SQL

```
--  File:     IDXCUSCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Creates clustered index on customer table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
```

```
select "Start date:", convert(varchar(30),@startdate,9)

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

create unique clustered index customer_c1 on customer(c_w_id, c_d_id,
c_id)
       on MSSQL70_cs_fg

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

go
```

IDXCUSNC.SQL

```
--  File:     IDXCUSNC.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Creates non-clustered index on customer table


use tpcc
go

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

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

create unique nonclustered index customer_nc1 on customer(c_w_id, c_d_id,
c_last, c_first, c_id)
       on MSSQL70_cs_fg

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

go
```

IDXDISCL.SQL

```
--  File:     IDXDISCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Creates clustered index on district table


use tpcc
go

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

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

create unique clustered index  district_c1 on district(d_w_id, d_id)
       with fillfactor=100 on MSSQL70_misc_fg

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

go
```

IDXITMCL.SQL

```
--  File:     IDXITMCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Creates clustered index on item table


use tpcc
go

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

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

create unique clustered index item_c1 on item(i_id)
       on MSSQL70_misc_fg

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

go
```

IDXNODCL.SQL

```
--  File:     IDXNODCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Creates clustered index on new_order table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
```

```
select "Start date:", convert(varchar(30),@startdate,9)

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

create unique clustered index new_order_c1 on new_order(no_w_id, no_d_id,
no_o_id)
       on MSSQL70_misc_fg

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

go
```

IDXODLCL.SQL

```
--  File:     IDXODLCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Creates clustered index on order_line table


use tpcc
go

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

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

create unique clustered index order_line_c1 on order_line(ol_w_id,
ol_d_id, ol_o_id, ol_number)
       on MSSQL70_misc_fg

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

go
```

IDXORDCL.SQL

```
--  File:     IDXORDCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Creates clustered index on orders table


use tpcc
go

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

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

create unique clustered index orders_c1 on orders(o_w_id, o_d_id, o_id)
       on MSSQL70_misc_fg

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

go
```

IDXORDNC.SQL

```
--  File:     IDXORDNC.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  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 unique nonclustered index orders_nc1 on orders(o_w_id, o_d_id,
o_c_id, o_id)
       with pad_index, fillfactor=90
       on MSSQL70_misc_fg

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

go
```

IDXSTKCL.SQL

```
--  File:     IDXSTKCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Creates clustered index on stock table


use tpcc
go

declare @startdate datetime
```

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

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

create unique clustered index stock_c1 on stock(s_i_id, s_w_id)
       on MSSQL70_cs_fg

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

go
```

IDXWARCL.SQL

```
--  File:     IDXWARCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Creates clustered index on warehouse table


use tpcc
go

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

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

create unique clustered index warehouse_c1 on warehouse(w_id)
       with fillfactor=100 on MSSQL70_misc_fg

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

go
```

TABLES.SQL

```
--  File:     TABLES.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Creates TPC-C tables


use tpcc
go

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

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


--
--  Create new tables
--

create table warehouse
(
       w_id                            smallint,
       w_name                          char(10),
       w_street_1                      char(20),
       w_street_2                      char(20),
       w_city                          char(20),
       w_state                         char(2),
       w_zip                           char(9),
       w_tax                           numeric(4,4),
       w_ytd                           numeric(12,2)
) on MSSQL70_misc_fg
go

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

```
) on MSSQL70_misc_fg
go

create table customer
(
       c_id                          int,
       c_d_id                        tinyint,
       c_w_id                        smallint,
       c_first                       char(16),
       c_middle                      char(2),
       c_last                        char(16),
       c_street_1                    char(20),
       c_street_2                    char(20),
       c_city                        char(20),
       c_state                       char(2),
       c_zip                         char(9),
       c_phone                       char(16),
       c_since                       datetime,
       c_credit                      char(2),
       c_credit_lim          numeric(12,2),
       c_discount                    numeric(4,4),
       c_balance                     numeric(12,2),
       c_ytd_payment         numeric(12,2),
       c_payment_cnt         smallint,
       c_delivery_cnt        smallint,
       c_data                        char(500)
) on MSSQL70_cs_fg
go

create table history
(
       h_c_id                        int,
       h_c_d_id                      tinyint,
       h_c_w_id                      smallint,
       h_d_id                        tinyint,
       h_w_id                        smallint,
       h_date                        datetime,
       h_amount                      numeric(6,2),
       h_data                        char(24)
) on MSSQL70_misc_fg
go

create table new_order
(
       no_o_id                       int,
       no_d_id                       tinyint,
       no_w_id                       smallint
) on MSSQL70_misc_fg
go

create table orders
(
       o_id                          int,
       o_d_id                        tinyint,
       o_w_id                        smallint,
       o_c_id                        int,
       o_entry_d                     datetime,
       o_carrier_id          tinyint,
       o_ol_cnt                      tinyint,
       o_all_local                   tinyint
) on MSSQL70_misc_fg
go

create table order_line
(
       ol_o_id                       int,
       ol_d_id                       tinyint,
       ol_w_id                       smallint,
       ol_number                     tinyint,
       ol_i_id                       int,
       ol_supply_w_id        smallint,
       ol_delivery_d         datetime,
       ol_quantity                   smallint,
       ol_amount                     numeric(6,2),
       ol_dist_info          char(24)
) on MSSQL70_misc_fg
go

create table item
(
       i_id                          int,
       i_im_id                       int,
       i_name                        char(24),
       i_price                       numeric(5,2),
       i_data                        char(50)
) on MSSQL70_misc_fg
go

create table stock
(
       s_i_id                        int,
       s_w_id                        smallint,
       s_quantity                    smallint,
       s_dist_01                     char(24),
       s_dist_02                     char(24),
       s_dist_03                     char(24),
       s_dist_04                     char(24),
       s_dist_05                     char(24),
       s_dist_06                     char(24),
       s_dist_07                     char(24),
       s_dist_08                     char(24),
       s_dist_09                     char(24),
       s_dist_10                     char(24),
       s_ytd                         int,
       s_order_cnt                   smallint,
       s_remote_cnt          smallint,
       s_data                        char(50)
) on MSSQL70_cs_fg
go
```

Stored Procedures

DELIVERY.SQL

```
--  File:     DELIVERY.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20.000
--            Copyright Microsoft, 1999
--  Purpose:  Creates delivery transaction stored procedure
```

```
--
--      Interface Level: 4.10.000

use tpcc
go

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

create proc tpcc_delivery     @w_id           smallint,

                              @o_carrier_id   smallint
as

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

select @d_id = 0

begin tran d

        while (@d_id < 10)
        begin

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

                select  top 1
                        @o_id   = no_o_id
                from    new_order (serializable updlock)
                where   no_w_id = @w_id and
                        no_d_id = @d_id
                order   by no_o_id asc

                if (@@rowcount <> 0)
                begin

--  claim the order for this district

                        delete  new_order
                        where   no_w_id = @w_id and
                                no_d_id = @d_id and
                                no_o_id = @o_id

--  set carrier_id on this order (and get customer id)

                        update  orders
                        set     o_carrier_id   = @o_carrier_id,
                                @c_id          = o_c_id
                        where   o_w_id         = @w_id and
                                o_d_id         = @d_id and
                                o_id           = @o_id

--  set date in all lineitems for this order (and sum amounts)

                        update  order_line
                        set     ol_delivery_d  = getdate(),
                                @total         = @total + ol_amount
                        where   ol_w_id        = @w_id and
                                ol_d_id        = @d_id and
                                ol_o_id        = @o_id

--  accummulate lineitem amounts for this order into customer

                        update  customer
                        set     c_balance      = c_balance + @total,
                                c_delivery_cnt = c_delivery_cnt + 1

                        where   c_w_id         = @w_id and
                                c_d_id         = @d_id and
                                c_id           = @c_id

                end

        select @oid1 = case @d_id when  1  then @o_id else @oid1 end,
               @oid2 = case @d_id when  2  then @o_id else @oid2 end,
               @oid3 = case @d_id when  3  then @o_id else @oid3 end,
               @oid4 = case @d_id when  4  then @o_id else @oid4 end,
               @oid5 = case @d_id when  5  then @o_id else @oid5 end,
               @oid6 = case @d_id when  6  then @o_id else @oid6 end,
               @oid7 = case @d_id when  7  then @o_id else @oid7 end,
               @oid8 = case @d_id when  8  then @o_id else @oid8 end,
               @oid9 = case @d_id when  9  then @o_id else @oid9 end,
               @oid10 = case @d_id when 10 then @o_id else @oid10 end

    end

commit tran d

-- return delivery data to client

select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

go
```

NEWORD.SQL

```
--  File:     NEWORD.SQL
```

```
--            Microsoft TPC-C Benchmark Kit Ver. 4.20.000
--            Copyright Microsoft, 1999
--  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
```

ORDSTAT.SQL

```
--  File:     ORDSTAT.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20.000
--            Copyright Microsoft, 1999
--  Purpose:  Creates order status transaction stored procedure
--
--      Interface Level: 4.10.000

use tpcc
go

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

create proc tpcc_orderstatus  @w_id   smallint,
                              @d_id   tinyint,

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

as

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

begin tran o

if (@c_id = 0)
        begin

--  get customer id and info using last name

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

                set     rowcount @cnt

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

                set     rowcount 0
        end

        else

        begin

--  get customer info if by id

                select  @c_balance      = c_balance,
                        @c_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
```

PAYMENTS.SQL

```
--  File:     PAYMENT.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20.000
--            Copyright Microsoft, 1999
--  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. 4.20.000
--            Copyright Microsoft, 1999
--  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. 4.20.000
--            Copyright Microsoft, 1999
--  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
```

Loader Source

GETARGS.C

```
//      File:           GETARGS.C
//                              Microsoft TPC-C Kit Ver. 4.20
//                              Copyright Microsoft, 1996, 1997, 1998, 1999
//      Purpose:        Source file for command line processing


// Includes
#include "tpcc.h"

//===========================================================================
```

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

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

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

    /* init args struct with some useful values */
    pargs->server                    =  SERVER;
    pargs->user                      =  USER;
    pargs->password                  =  PASSWORD;
    pargs->database                  =  DATABASE;
    pargs->batch                     =  BATCH;
    pargs->num_warehouses            =  UNDEF;
       pargs->tables_all                    =  TRUE;
       pargs->table_item                    =  FALSE;
       pargs->table_warehouse        =  FALSE;
       pargs->table_customer         =  FALSE;
       pargs->table_orders                  =  FALSE;
       pargs->loader_res_file               =  LOADER_RES_FILE;
       pargs->pack_size                     =  DEFLDPACKSIZE;
       pargs->starting_warehouse     =  DEF_STARTING_WAREHOUSE;
       pargs->build_index                   =  BUILD_INDEX;
       pargs->index_order                   =  INDEX_ORDER;
       pargs->index_script_path      =  INDEX_SCRIPT_PATH;
       pargs->scale_down                    =  SCALE_DOWN;

    /* check for zero command line args */
    if ( argc == 1 )
               GetArgsLoaderUsage();

    for (i = 1; i < argc; ++i)
    {
               if (argv[i][0] != '-' && argv[i][0] != '/')
               {
               printf("\nUnrecognized command");
               GetArgsLoaderUsage();
               exit(1);
               }

               ptr = argv[i];

               switch (ptr[1])
               {
               case 'h':     /* Fall throught */
               case 'H':
                              GetArgsLoaderUsage();
                              break;

               case 'D':
                              pargs->database = ptr+2;
                              break;
               case 'P':
                              pargs->password = ptr+2;
                              break;
               case 'S':
                              pargs->server = ptr+2;
                              break;
               case 'U':
                              pargs->user = ptr+2;
                              break;
               case 'b':
                              pargs->batch = atol(ptr+2);
                              break;
               case 'W':
                              pargs->num_warehouses = atol(ptr+2);
                              break;
               case 's':
                              pargs->starting_warehouse = atol(ptr+2);
                              break;
               case 't':
                              {
                                     pargs->tables_all = FALSE;
                                     if (strcmp(ptr+2,"item") == 0)
                                            pargs->table_item = TRUE;
                                     else if (strcmp(ptr+2,"warehouse")
== 0)
                                            pargs->table_warehouse =
TRUE;
                                     else if (strcmp(ptr+2,"customer") ==
0)
                                            pargs->table_customer = TRUE;
                                     else if (strcmp(ptr+2,"orders") ==
0)
                                            pargs->table_orders = TRUE;
                                     else
                                     {
                                     printf("\nUnrecognized command");
                                     GetArgsLoaderUsage();
                                     exit(1);
                                     }

                                     break;
                              }
                       case 'f':
                              pargs->loader_res_file = ptr+2;
                              break;
                       case 'p':
                              pargs->pack_size = atol(ptr+2);
                              break;
                       case 'i':
                              pargs->build_index = atol(ptr+2);
                              break;
```

```
                        case 'o':
                                pargs->index_order = atol(ptr+2);
                                break;

                        case 'c':
                                pargs->scale_down = atol(ptr+2);
                                break;

                case 'd':
                                pargs->index_script_path = ptr+2;
                                break;

                default:
                        GetArgsLoaderUsage();
                        exit(-1);
                        break;
                }
    }

    /* check for required args */
    if (pargs->num_warehouses == UNDEF )
    {
                printf("Number of Warehouses is required\n");
                exit(-2);
    }

    return;
}


//===============================================================================
//
// Function name: GetArgsLoaderUsage
//
//===============================================================================

void GetArgsLoaderUsage()
{

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


    printf("TPCCLDR:\n\n");
       printf("Parameter
Default\n");
       printf("-------------------------------------------------------------
----\n");
    printf("-W Number of Warehouses to Load                      Required
\n");
    printf("-S Server                                            %s\n",
SERVER);
    printf("-U Username                                          %s\n",
USER);
    printf("-P Password                                          %s\n",
PASSWORD);
    printf("-D Database                                          %s\n",
DATABASE);
       printf("-b Batch Size
%ld\n", (long) BATCH);
       printf("-p TDS packet size
%ld\n", (long) DEFLDPACKSIZE);
       printf("-f Loader Results Output Filename
%s\n", LOADER_RES_FILE);
       printf("-s Starting Warehouse
%ld\n", (long) DEF_STARTING_WAREHOUSE);
       printf("-i Build Option (data = 0, data and index = 1)
%ld\n", (long) BUILD_INDEX);
       printf("-o Cluster Index Build Order (before = 1, after = 0)
%ld\n", (long) INDEX_ORDER);
       printf("-c Build Scaled Database (normal = 0, tiny = 1)
%ld\n", (long) SCALE_DOWN);
       printf("-d Index Script Path
%s\n", INDEX_SCRIPT_PATH);
       printf("-t Table to Load                                      all
tables \n");
    printf("   [item|warehouse|customer|orders]\n");
    printf("   Notes: \n");
    printf("     - the '-t' parameter may be included multiple times to
\n");
    printf("       specify multiple tables to be loaded \n");
    printf("     - 'item' loads ITEM table \n");
    printf("     - 'warehouse' loads WAREHOUSE, DISTRICT, and STOCK tables
\n");
    printf("     - 'customer' loads CUSTOMER and HISTORY tables \n");
    printf("     - 'orders' load NEW-ORDER, ORDERS, ORDER-LINE tables
\n");

       printf("\nNote:  Command line switches are case sensitive.\n");

    exit(0);
}
```

RANDOM.C

```
//      File:          RANDOM.C
//                             Microsoft TPC-C Kit Ver. 4.20
//                             Copyright Microsoft, 1996, 1997, 1998, 1999
//      Purpose:       Random number generation routines for database
loader


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

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

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

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

/*****************************************************************************
*****
* seed - load the Seed value used in irand and drand.  Should be used
before  *
*       first call to irand or drand.
*
******************************************************************************
****/

void seed(long val)
{

#ifdef DEBUG
    printf("[%ld]DBG: Entering seed()...\n", (int) GetCurrentThreadId());
       printf("Old Seed %ld New Seed %ld\n",Seed, val);
#endif

    if ( val < 0 )
            val = abs(val);

    Seed = val;

}


/*****************************************************************************
****
*
*
* irand - returns a 32 bit integer pseudo random number with a period of
*
*       1 to 2 ^ 32 - 1.
*
*
*
* parameters:
*
*       none.
*
*
*
* returns:
*
*       32 bit integer - defined as long ( see above ).
*
*
*
* side effects:
*
*       seed get recomputed.
*
******************************************************************************
***/

long irand()
{
    register long    s;      /* copy of seed */
    register long    test;   /* test flag */
    register long    hi;     /* tmp value for speed */
    register long    lo;     /* tmp value for speed */

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

    s = Seed;
    hi = s / Q;
    lo = s % Q;

    test = A * lo - R * hi;
    if ( test > 0 )
                Seed = test;
    else
                Seed = test + M;

    return( Seed );
}


/*****************************************************************************
****
*
*
* drand - returns a double pseudo random number between 0.0 and 1.0.
*
*       See irand.
*
******************************************************************************
***/
double drand()
{
```

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

    return( (double)irand() / 2147483647.0);
}


//==============================================================================
// Function   : RandomNumber
//
// Description:
//==============================================================================
long RandomNumber(long lower, long upper)
{
    long rand_num;

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

        if ( upper == lower )  /* pgd 08-13-96 perf enhancement */
                return lower;

        upper++;

    if ( upper <= lower )
                rand_num = upper;
    else
                rand_num = lower + irand() % (upper - lower); /* pgd 08-13-
96 perf enhancement */


#ifdef DEBUG
    printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                              (int) GetCurrentThreadId(), lower, upper,
rand_num);
#endif


    return rand_num;
}


#if 0

//Orginal code pgd 08/13/96

long RandomNumber(long lower,
                                 long upper)
{
    long rand_num;

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

        upper++;

    if ((upper <= lower))
                rand_num = upper;
    else
                rand_num = lower + irand() % ((upper > lower) ? upper -
lower : upper);


#ifdef DEBUG
    printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                                (int) GetCurrentThreadId(), lower, upper,
rand_num);
#endif


    return rand_num;
}
#endif

//==============================================================================
// Function   : NURand
//
// Description:
//==============================================================================
long NURand(int iConst,
            long x,
            long y,
            long C)
{
    long rand_num;

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

    rand_num = (((RandomNumber(0,iConst) | RandomNumber(x,y)) + C) % (y-
x+1))+x;

#ifdef DEBUG
    printf("[%ld]DBG: NURand: num = %d\n", (int) GetCurrentThreadId(),
rand_num);
#endif

    return rand_num;
}
```

STRINGS.C

```
//      File:           STRINGS.C
//                              Microsoft TPC-C Kit Ver. 4.20
//                              Copyright Microsoft, 1996, 1997, 1998, 1999
//      Purpose:        Source file for database loader string functions


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

```
//==============================================================================
//
// Function name: MakeAddress
//
//==============================================================================

void MakeAddress(char *street_1,
                                char *street_2,
                                char *city,
                                char *state,
                                char *zip)
{

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

    MakeAlphaString (10, 20, ADDRESS_LEN, street_1);
    MakeAlphaString (10, 20, ADDRESS_LEN, street_2);
    MakeAlphaString (10, 20, ADDRESS_LEN, city);
    MakeAlphaString ( 2,  2, STATE_LEN, state);
    MakeZipNumberString( 9,  9, ZIP_LEN, zip);

#ifdef DEBUG
    printf("[%ld]DBG: MakeAddress: street_1: %s, street_2: %s, city: %s,
state: %s, zip: %s\n",
                        (int) GetCurrentThreadId(), street_1, street_2,
city, state, zip);
#endif

    return;

}


//==============================================================================
//
// Function name: LastName
//
//==============================================================================

void LastName(int  num,
                      char *name)
{
    static char *n[] =
    {
                "BAR" , "OUGHT", "ABLE" , "PRI"   , "PRES",
                "ESE" , "ANTI" , "CALLY", "ATION", "EING"
    };

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

    if ((num >= 0) && (num < 1000))
    {
                strcpy(name, n[(num/100)%10]);
                strcat(name, n[(num/10)%10]);
                strcat(name, n[(num/1)%10]);
                if (strlen(name) < LAST_NAME_LEN)
                {
                        PaddString(LAST_NAME_LEN, name);
                }

    }
    else
    {
                printf("\nError in LastName()... num <%ld> out of range
(0,999)\n", num);
                exit(-1);
    }


#ifdef DEBUG
    printf("[%ld]DBG: LastName: num = [%d] ==> [%d][%d][%d]\n",
                        (int) GetCurrentThreadId(), num, num/100,
(num/10)%10, num%10);
        printf("[%ld]DBG: LastName: String = %s\n", (int)
GetCurrentThreadId(), name);
#endif

    return;

}


//==============================================================================
//
// Function name: MakeAlphaString
//
//==============================================================================

//philipdu 08/13/96 Changed MakeAlphaString to use A-Z, a-z, and 0-9 in
//accordance with spec see below:
//The spec says:
//4.3.2.2      The notation random a-string [x .. y]
//(respectively, n-string [x .. y]) represents a string of random
alphanumeric
//(respectively, numeric) characters of a random length of minimum x,
maximum y,
//and mean (y+x)/2. Alphanumerics are A..Z, a..z, and 0..9.  The only
other
//requirement is that the character set used "must be able to represent a
minimum
//of 128 different characters".  We are using 8-bit chars, so this is a
non issue.
//It is completely unreasonable to stuff non-printing chars into the text
fields.
//-CLevine 08/13/96

int MakeAlphaString( int  x, int  y, int  z, char *str)
{
        int             len;
        int             i;
    char    cc = 'a';
        static  char chArray[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
        static  int     chArrayMax = 61;

#ifdef DEBUG
```

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

        len= RandomNumber(x, y);

        for (i=0; i<len; i++)
        {
                cc = chArray[RandomNumber(0, chArrayMax)];
         str[i] = cc;
    }
        if ( len < z )
                memset(str+len, ' ', z - len);
        str[len] = 0;

    return len;
}


//=========================================================================
//
// Function name: MakeOriginalAlphaString
//
//=========================================================================

int MakeOriginalAlphaString(int  x,
                                                int  y,
                                                   int  z,
                                                char *str,
                                              int  percent)
{
    int        len;
    int        val;
    int        start;

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

    // verify prercentage is valid
    if ((percent < 0) || (percent > 100))
    {
                printf("MakeOrigianlAlphaString: Invalid percentage: %d\n",
percent);
                exit(-1);
    }

    // verify string is at least 8 chars in length
    if ((x + y) <= 8)
    {
                printf("MakeOriginalAlphaString: string length must be >=
8\n");
                exit(-1);
    }

    // Make Alpha String
    len = MakeAlphaString(x,y, z, str);

    val = RandomNumber(1,100);
    if (val <= percent)
    {
                start = RandomNumber(0, len - 8);
                strncpy(str + start, "ORIGINAL", 8);
    }


#ifdef DEBUG
    printf("[%ld]DBG: MakeOriginalAlphaString: : %s\n",
                        (int) GetCurrentThreadId(), str);
#endif

    return strlen(str);
}


//=========================================================================
//
// Function name: MakeNumberString
//
//=========================================================================
int MakeNumberString(int  x, int  y, int  z, char *str)
{
        char tmp[16];

        //MakeNumberString is always called MakeZipNumberString(16, 16,
16, string)

        memset(str, '0', 16);
        itoa(RandomNumber(0, 99999999), tmp, 10);
        memcpy(str, tmp, strlen(tmp));

        itoa(RandomNumber(0, 99999999), tmp, 10);
        memcpy(str+8, tmp, strlen(tmp));

        str[16] = 0;

    return 16;
}


//=========================================================================
//
// Function name: MakeZipNumberString
//
//=========================================================================
int MakeZipNumberString(int  x, int  y, int  z, char *str)
{
        char tmp[16];

        //MakeZipNumberString is always called MakeZipNumberString(9, 9,
9, string)

        strcpy(str, "000011111");

        itoa(RandomNumber(0, 9999), tmp, 10);
        memcpy(str, tmp, strlen(tmp));

    return 9;
}
```

```
//=====================================================================
//
// Function name: InitString
//
//=====================================================================
void InitString(char *str, int len)
{
#ifdef DEBUG
    printf("[%ld]DBG: Entering InitString()\n", (int)
GetCurrentThreadId());
#endif

        memset(str, ' ', len);
    str[len] = 0;
}


//=====================================================================
// Function name: InitAddress
//
// Description:
//
//=====================================================================

void InitAddress(char *street_1, char *street_2, char *city, char *state,
char *zip)
{
        memset(street_1, ' ', ADDRESS_LEN+1);
        memset(street_2, ' ', ADDRESS_LEN+1);
        memset(city, ' ', ADDRESS_LEN+1);

    street_1[ADDRESS_LEN+1] = 0;
    street_2[ADDRESS_LEN+1] = 0;
    city[ADDRESS_LEN+1]     = 0;

        memset(state, ' ', STATE_LEN+1);
    state[STATE_LEN+1] = 0;

        memset(zip, ' ', ZIP_LEN+1);
    zip[ZIP_LEN+1] = 0;
}


//=====================================================================
//
// Function name: PaddString
//
//=====================================================================

void PaddString(int max, char *name)
{
        int             len;

        len = strlen(name);
        if ( len < max )
                memset(name+len, ' ', max - len);
        name[max] = 0;

        return;
}
```

TIME.C

```
//      File:           TIME.C
//                              Microsoft TPC-C Kit Ver. 4.20
//                              Copyright Microsoft, 1996, 1997, 1998, 1999
//      Purpose:        Source file for time functions


// Includes
#include "tpcc.h"

// Globals
static long start_sec;


//=====================================================================
//
// Function name: TimeNow
//
//=====================================================================

long TimeNow()
{
   long         time_now;
   struct       _timeb el_time;

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

   _ftime(&el_time);

   time_now = ((el_time.time - start_sec) * 1000) + el_time.millitm;

   return time_now;
}
```

TPCC.H

```
//      File:           TPCC.H
//                              Microsoft TPC-C Kit Ver. 4.20
//                              Copyright Microsoft, 1996, 1997, 1998, 1999
//      Purpose:        Header file for TPC-C database loader


// Build number of TPC Benchmark Kit
#define TPCKIT_VER      "4.20"

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

```
#include <sys\types.h>

// ODBC headers
#include <sql.h>
#include <sqlext.h>
#include <odbcss.h>

// General constants
#define MILLI                          1000
#define FALSE                                0
#define TRUE                                 1
#define UNDEF                                   -1
#define MINPRINTASCII                   32
#define MAXPRINTASCII                   126

// Default environment constants
#define SERVER                               ""
#define DATABASE                             "tpcc"
#define USER                                 "sa"
#define PASSWORD                          ""

// Default loader arguments
#define BATCH                          10000
#define DEFLDPACKSIZE                  32768
#define LOADER_RES_FILE                  "logs\\load.out"
#define LOADER_NURAND_C                    123
#define DEF_STARTING_WAREHOUSE               1
#define BUILD_INDEX                                 1      // build both
data and indexes
#define INDEX_ORDER                                 1      // build
indexes before load
#define SCALE_DOWN                                  0   // build a normal
scale database
#define INDEX_SCRIPT_PATH     "scripts"

typedef struct
{
    char                           *server;
    char                           *database;
    char                           *user;
    char                           *password;
        BOOL                       tables_all;                  // set
if loading all tables
        BOOL                       table_item;                  // set
if loading ITEM table specifically
        BOOL                       table_warehouse;       // set if
loading WAREHOUSE, DISTRICT, and STOCK
        BOOL                       table_customer;              // set
if loading CUSTOMER and HISTORY
        BOOL                       table_orders;          // set if
loading NEW-ORDER, ORDERS, ORDER-LINE
    long                           num_warehouses;
    long                           batch;
    long                           verbose;
        long                       pack_size;
        char                       *loader_res_file;
        char                       *synch_servername;
        long                       case_sensitivity;
        long                       starting_warehouse;
        long                       build_index;
        long                       index_order;
        long                       scale_down;
        char                       *index_script_path;
} TPCCLDR_ARGS;

// String length constants
        #define SERVER_NAME_LEN            20
        #define DATABASE_NAME_LEN          20
        #define USER_NAME_LEN              20
        #define PASSWORD_LEN               20
        #define TABLE_NAME_LEN             20
        #define I_DATA_LEN                 50
        #define I_NAME_LEN                 24
        #define BRAND_LEN                   1
        #define LAST_NAME_LEN              16
        #define W_NAME_LEN                 10
        #define ADDRESS_LEN                20
        #define STATE_LEN                   2
        #define ZIP_LEN                     9
        #define S_DIST_LEN                 24
        #define S_DATA_LEN                 50
        #define D_NAME_LEN                 10
        #define FIRST_NAME_LEN             16
        #define MIDDLE_NAME_LEN             2
        #define PHONE_LEN                  16
        #define CREDIT_LEN                  2
        #define C_DATA_LEN                500
        #define H_DATA_LEN                 24
        #define DIST_INFO_LEN              24
        #define MAX_OL_NEW_ORDER_ITEMS     15
        #define MAX_OL_ORDER_STATUS_ITEMS  15
        #define STATUS_LEN                 25
        #define OL_DIST_INFO_LEN                  24
        #define C_SINCE_LEN                                 23
        #define H_DATE_LEN                                  23
        #define OL_DELIVERY_D_LEN                 23
        #define O_ENTRY_D_LEN              23


// Functions in random.c
void    seed();
long    irand();
double  drand();
void    WUCreate();
short   WURand();
long    RandomNumber(long lower, long upper);

// Functions in getargs.c;
void    GetArgsLoader();
void    GetArgsLoaderUsage();

// Functions in time.c
long    TimeNow();

// Functions in strings.c
void    MakeAddress();
void    LastName();
int     MakeAlphaString();
int     MakeOriginalAlphaString();
int     MakeNumberString();
int     MakeZipNumberString();
void    InitString();
```

```
void    InitAddress();
void    PaddString();
```

TPCCLDR.C

```
//      File:           TPCCLDR.C
//                              Microsoft TPC-C Kit Ver. 4.20
//                              Copyright Microsoft, 1996, 1997, 1998, 1999
//      Purpose:        Source file for TPC-C database loader


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

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

// Functions declarations

void HandleErrorDBC (SQLHDBC  hdbc1);

void CheckSQL();
void CheckDataBase();

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

void Stock();
void District();

void LoadCustomer();
void CustomerBufInit();
void CustomerBufLoad();
void LoadCustomerTable();
void LoadHistoryTable();

void LoadOrders();
void OrdersBufInit();
void OrdersBufLoad();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void GetPermutation();
void CheckForCommit();
void OpenConnections();
void BuildIndex();
void FormatDate ();

// Shared memory structures


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

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

typedef struct
{
    long                                c_id;
    short                               c_d_id;
    short                               c_w_id;
       char                             c_first[FIRST_NAME_LEN+1];
    char                                c_middle[MIDDLE_NAME_LEN+1];
    char                                c_last[LAST_NAME_LEN+1];
    char                                c_street_1[ADDRESS_LEN+1];
    char                                c_street_2[ADDRESS_LEN+1];
    char                                c_city[ADDRESS_LEN+1];
    char                                c_state[STATE_LEN+1];
    char                                c_zip[ZIP_LEN+1];
    char                                c_phone[PHONE_LEN+1];
    char                                c_credit[CREDIT_LEN+1];
    double                              c_credit_lim;
    double                              c_discount;
//  fix to avoid ODBC float to numeric conversion problem.
//  double                              c_balance;
       char                             c_balance[6];

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

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

typedef struct
```

```
{
    long                time_start;
} LOADER_TIME_STRUCT;


// Global variables

char    szLastError[300];

HENV    henv;

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

HDBC    o_hdbc3;                                        // for ORDER-LINE

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

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


TPCCLDR_ARGS    *aptr, args;


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

int main(int  argc, char **argv)
{
       DWORD           dwThreadID[MAX_MAIN_THREADS];
       HANDLE          hThread[MAX_MAIN_THREADS];
       FILE            *fLoader;
       char            buffer[255];
       int                             i;

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

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

    // process command line arguments

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

       // verify correct SQL Server version in use
       // you must be using SQL Server 7.00.623 or better to load

       CheckSQL();

       // verify database and tables exist before attempting to load

       CheckDataBase();

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

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

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

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

```
                max_items = MAXITEMS;
                customers_per_district = CUSTOMERS_PER_DISTRICT;
                orders_per_district = ORDERS_PER_DISTRICT;
                first_new_order = 2100;
                last_new_order  = 3000;
        }


        // open connections to SQL Server

        OpenConnections();

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

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

    // start loading data

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

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

        main_time_start = (TimeNow() / MILLI);

        // start parallel load threads


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

                hThread[0] = CreateThread(NULL,
                                                                0,

(LPTHREAD_START_ROUTINE) LoadItem,

                                                                NULL,
                                                                0,


&dwThreadID[0]);

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

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

                hThread[1] = CreateThread(NULL,
                                                                0,

(LPTHREAD_START_ROUTINE) LoadWarehouse,

                                                                NULL,
                                                                0,


&dwThreadID[1]);

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

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

                hThread[2] = CreateThread(NULL,
                                                                0,

(LPTHREAD_START_ROUTINE) LoadCustomer,

                                                                NULL,
                                                                0,


&dwThreadID[2]);

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

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

                hThread[3] = CreateThread(NULL,
                                                                0,

(LPTHREAD_START_ROUTINE) LoadOrders,

                                                                NULL,
                                                                0,


&dwThreadID[3]);

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

        // Wait for threads to finish...
        for (i=0; i<MAX_MAIN_THREADS; i++)
```

```
        {
                if (hThread[i] != NULL)
                {
                        WaitForSingleObject( hThread[i], INFINITE );
                        CloseHandle(hThread[i]);
                        hThread[i] = NULL;
                }
        }

        main_time_end = (TimeNow() / MILLI);

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

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

        fclose(fLoader);

        SQLFreeEnv(henv);

    exit(0);

        return 0;
}


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

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

        // Seed with unique number
        seed(1);

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

        // if build index before load
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
                BuildIndex("idxitmcl");

        InitString(i_name, I_NAME_LEN+1);
        InitString(i_data, I_DATA_LEN+1);

        sprintf(name, "%s..%s", aptr->database, "item");

        rc = bcp_init(i_hdbc1, name, NULL, "logs\\item.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (i_id), ROWS_PER_BATCH =
100000");
                rc = bcp_control(i_hdbc1, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);
        }

        rc = bcp_bind(i_hdbc1, (BYTE *) &i_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT4, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        rc = bcp_bind(i_hdbc1, (BYTE *) i_name, 0, I_NAME_LEN, NULL, 0, 0,
3);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        rc = bcp_bind(i_hdbc1, (BYTE *) &i_price, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0, I_DATA_LEN, NULL, 0, 0,
5);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        time_start = (TimeNow() / MILLI);

        item_rows_loaded = 0;

        for (i_id = 1; i_id <= max_items; i_id++)
        {
                i_im_id = RandomNumber(1L, 10000L);

                MakeAlphaString(14, 24, I_NAME_LEN, i_name);

                i_price = ((float) RandomNumber(100L, 10000L))/100.0;

                MakeOriginalAlphaString(26, 50, I_DATA_LEN, i_data, 10);

                rc = bcp_sendrow(i_hdbc1);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

                item_rows_loaded++;
                CheckForCommit(i_hdbc1, i_hstmt1, item_rows_loaded, "item",
&time_start);
        }
```

```
        rcint = bcp_done(i_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(i_hdbc1);

        printf("Finished loading item table.\n");

        SQLFreeStmt(i_hstmt1, SQL_DROP);
        SQLDisconnect(i_hdbc1);
        SQLFreeConnect(i_hdbc1);

        // if build index after load
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxitmcl");

}


//============================================================================
=====
//
// Function   : LoadWarehouse
//
// Loads WAREHOUSE table and loads Stock and District as Warehouses are
created
//
//============================================================================
=====

void LoadWarehouse()
{
    short      w_id;
    char       w_name[W_NAME_LEN+1];
    char       w_street_1[ADDRESS_LEN+1];
    char       w_street_2[ADDRESS_LEN+1];
    char       w_city[ADDRESS_LEN+1];
    char       w_state[STATE_LEN+1];
    char       w_zip[ZIP_LEN+1];
    double     w_tax;
    double     w_ytd;
        char    name[20];
        long    time_start;
        RETCODE rc;
        DBINT   rcint;
        char    bcphint[128];


        // Seed with unique number
        seed(2);

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

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
                BuildIndex("idxwarcl");

        InitString(w_name, W_NAME_LEN+1);
        InitAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

        sprintf(name, "%s..%s", aptr->database, "warehouse");

        rc = bcp_init(w_hdbc1, name, NULL, "logs\\whouse.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (w_id), ROWS_PER_BATCH =
%d", aptr->num_warehouses);
                rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);
        }

        rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0, W_NAME_LEN, NULL, 0, 0,
2);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_street_1, 0, ADDRESS_LEN, NULL,
0, 0, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_street_2, 0, ADDRESS_LEN, NULL,
0, 0, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_city, 0, ADDRESS_LEN, NULL, 0,
0, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_state, 0, STATE_LEN, NULL, 0, 0,
6);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_zip, 0, ZIP_LEN, NULL, 0, 0, 7);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0, SQL_VARLEN_DATA, NULL,
0, SQLFLT8, 8);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0, SQL_VARLEN_DATA, NULL,
0, SQLFLT8, 9);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        time_start = (TimeNow() / MILLI);

        warehouse_rows_loaded = 0;
```

```
        for (w_id = (short)aptr->starting_warehouse; w_id <= aptr-
>num_warehouses; w_id++)
        {
                MakeAlphaString(6,10, W_NAME_LEN, w_name);

                MakeAddress(w_street_1, w_street_2, w_city, w_state,
w_zip);

                w_tax = ((float) RandomNumber(0L,2000L))/10000.00;

                w_ytd = 300000.00;

                rc = bcp_sendrow(w_hdbc1);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                warehouse_rows_loaded++;
                CheckForCommit(w_hdbc1, i_hstmt1, warehouse_rows_loaded,
"warehouse", &time_start);
        }

        rcint = bcp_done(w_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(w_hdbc1);

        printf("Finished loading warehouse table.\n");

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxwarcl");

        stock_rows_loaded = 0;
        district_rows_loaded = 0;

        District();
        Stock();

}


//==============================================================================
//
// Function   : District
//
//==============================================================================

void District()
{
    short      d_id;
    short      d_w_id;
    char       d_name[D_NAME_LEN+1];
    char       d_street_1[ADDRESS_LEN+1];
    char       d_street_2[ADDRESS_LEN+1];
    char       d_city[ADDRESS_LEN+1];
    char       d_state[STATE_LEN+1];
    char       d_zip[ZIP_LEN+1];
    double     d_tax;
    double     d_ytd;
        char    name[20];
    long       d_next_o_id;
        long    time_start;
        int             w_id;
        RETCODE rc;
        DBINT   rcint;
        char    bcphint[128];

        // Seed with unique number
        seed(4);

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

        // build index before load
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
                BuildIndex("idxdiscl");

        InitString(d_name, D_NAME_LEN+1);
        InitAddress(d_street_1, d_street_2, d_city, d_state, d_zip);
        sprintf(name, "%s..%s", aptr->database, "district");

        rc = bcp_init(w_hdbc1, name, NULL, "logs\\district.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (d_w_id, d_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 10));
                rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);
        }

        rc = bcp_bind(w_hdbc1, (BYTE *) &d_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &d_w_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) d_name, 0, D_NAME_LEN, NULL, 0, 0,
3);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) d_street_1, 0, ADDRESS_LEN, NULL,
0, 0, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) d_street_2, 0, ADDRESS_LEN, NULL,
0, 0, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) d_city, 0, ADDRESS_LEN, NULL, 0,
0, 6);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
```

```
        rc = bcp_bind(w_hdbc1, (BYTE *) d_state, 0, STATE_LEN, NULL, 0, 0,
7);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) d_zip, 0, ZIP_LEN, NULL, 0, 0, 8);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &d_tax, 0, SQL_VARLEN_DATA, NULL,
0, SQLFLT8, 9);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &d_ytd, 0, SQL_VARLEN_DATA, NULL,
0, SQLFLT8, 10);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &d_next_o_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT4, 11);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        d_ytd  = 30000.0;

        d_next_o_id = orders_per_district+1;

        time_start = (TimeNow() / MILLI);

        for (w_id = aptr->starting_warehouse; w_id <= aptr-
>num_warehouses; w_id++)
        {
                d_w_id = w_id;

                for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
                {
                        MakeAlphaString(6,10,D_NAME_LEN, d_name);

                        MakeAddress(d_street_1, d_street_2, d_city,
d_state, d_zip);

                        d_tax = ((float) RandomNumber(0L,2000L))/10000.00;

                        rc = bcp_sendrow(w_hdbc1);
                        if (rc != SUCCEED)
                                HandleErrorDBC(w_hdbc1);

                        district_rows_loaded++;
                        CheckForCommit(w_hdbc1, w_hstmt1,
district_rows_loaded, "district", &time_start);
                }
    }

        rcint = bcp_done(w_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(w_hdbc1);

        printf("Finished loading district table.\n");

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxdiscl");

    return;
}


//================================================================================
//
// Function   : Stock
//
//================================================================================

void Stock()
{
    long       s_i_id;
    short      s_w_id;
    short      s_quantity;
    char       s_dist_01[S_DIST_LEN+1];
    char       s_dist_02[S_DIST_LEN+1];
    char       s_dist_03[S_DIST_LEN+1];
    char       s_dist_04[S_DIST_LEN+1];
    char       s_dist_05[S_DIST_LEN+1];
    char       s_dist_06[S_DIST_LEN+1];
    char       s_dist_07[S_DIST_LEN+1];
    char       s_dist_08[S_DIST_LEN+1];
    char       s_dist_09[S_DIST_LEN+1];
    char       s_dist_10[S_DIST_LEN+1];
    long       s_ytd;
    short      s_order_cnt;
    short      s_remote_cnt;
    char       s_data[S_DATA_LEN+1];
    short      len;
       char    name[20];
       long    time_start;
       RETCODE rc;
       DBINT   rcint;
       char    bcphint[128];

       // Seed with unique number
       seed(3);

       // if build index before load...
       if ((aptr->build_index == 1) && (aptr->index_order == 1))
              BuildIndex("idxstkcl");

       sprintf(name, "%s..%s", aptr->database, "stock");

       rc = bcp_init(w_hdbc1, name, NULL, "logs\\stock.err", DB_IN);
       if (rc != SUCCEED)
              HandleErrorDBC(w_hdbc1);

       if ((aptr->build_index == 1) && (aptr->index_order == 1))
       {
              sprintf(bcphint, "tablock, order (s_i_id, s_w_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 100000));
              rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
              if (rc != SUCCEED)
                     HandleErrorDBC(w_hdbc1);
       }
```

```
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &s_quantity, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01, 0, S_DIST_LEN, NULL, 0,
0, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02, 0, S_DIST_LEN, NULL, 0,
0, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03, 0, S_DIST_LEN, NULL, 0,
0, 6);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04, 0, S_DIST_LEN, NULL, 0,
0, 7);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05, 0, S_DIST_LEN, NULL, 0,
0, 8);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06, 0, S_DIST_LEN, NULL, 0,
0, 9);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07, 0, S_DIST_LEN, NULL, 0,
0, 10);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0, S_DIST_LEN, NULL, 0,
0, 11);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09, 0, S_DIST_LEN, NULL, 0,
0, 12);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0, S_DIST_LEN, NULL, 0,
0, 13);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 14);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &s_order_cnt, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 15);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &s_remote_cnt, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 16);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0, S_DATA_LEN, NULL, 0, 0,
17);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        s_ytd = s_order_cnt = s_remote_cnt = 0;

        time_start = (TimeNow() / MILLI);

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

        for (s_i_id=1; s_i_id <= max_items; s_i_id++)
        {

                for (s_w_id = (short)aptr->starting_warehouse; s_w_id <=
aptr->num_warehouses; s_w_id++)
                {

                        s_quantity = (short)RandomNumber(10L,100L);
                        len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_01);
                        len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_02);
                        len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_03);
                        len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_04);
                        len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_05);
                        len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_06);
                        len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_07);
                        len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_08);
                        len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_09);
                        len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_10);

                        len = MakeOriginalAlphaString(26,50, S_DATA_LEN,
s_data,10);

                        rc = bcp_sendrow(w_hdbc1);
                        if (rc != SUCCEED)
                                HandleErrorDBC(w_hdbc1);

                        stock_rows_loaded++;
                        CheckForCommit(w_hdbc1, w_hstmt1,
stock_rows_loaded, "stock", &time_start);
```

```
                }
        }

        rcint = bcp_done(w_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(w_hdbc1);

        printf("Finished loading stock table.\n");

        SQLFreeStmt(w_hstmt1, SQL_DROP);
        SQLDisconnect(w_hdbc1);
        SQLFreeConnect(w_hdbc1);

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxstkcl");

        return;
}


//============================================================================
//
// Function   : LoadCustomer
//
//============================================================================

void LoadCustomer()
{
        LOADER_TIME_STRUCT      customer_time_start;
        LOADER_TIME_STRUCT      history_time_start;
        short                           w_id;
    short                       d_id;
        DWORD                           dwThreadID[MAX_CUSTOMER_THREADS];
        HANDLE                          hThread[MAX_CUSTOMER_THREADS];
        char                            name[20];
        RETCODE                                 rc;
        DBINT                                   rcint;
        char                                    bcphint[128];
        char                                    cmd[256];
        // SQLRETURN                            rc_1;
        // SQLSMALLINT                          recnum, MsgLen;
        // SQLCHAR                                      SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
        // SQLINTEGER                           NativeError;

        // Seed with unique number
        seed(5);

        printf("Loading customer and history tables...\n");

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
                BuildIndex("idxcuscl");

        // Initialize bulk copy
        sprintf(name, "%s..%s", aptr->database, "customer");

        rc = bcp_init(c_hdbc1, name, NULL, "logs\\customer.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (c_w_id, c_d_id, c_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
                rc = bcp_control(c_hdbc1, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);
        }

        sprintf(name, "%s..%s", aptr->database, "history");

        rc = bcp_init(c_hdbc2, name, NULL, "logs\\history.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

        sprintf(bcphint, "tablock");
        rc = bcp_control(c_hdbc2, BCPHINTS, (void*) bcphint);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

        customer_rows_loaded    = 0;
        history_rows_loaded     = 0;

        CustomerBufInit();

        customer_time_start.time_start = (TimeNow() / MILLI);
        history_time_start.time_start = (TimeNow() / MILLI);

        for (w_id = (short)aptr->starting_warehouse; w_id <= aptr-
>num_warehouses; w_id++)
        {
                for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
                {
                        CustomerBufLoad(d_id, w_id);

                        // Start parallel loading threads here...

                        // Start customer table thread

                        printf("...Loading customer table for: d_id = %d,
w_id = %d\n", d_id, w_id);

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

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

```
                        // Start History table thread

                        printf("...Loading history table for: d_id = %d,
w_id = %d\n", d_id, w_id);

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

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

                        WaitForSingleObject( hThread[0], INFINITE );
                        WaitForSingleObject( hThread[1], INFINITE );

                        if (CloseHandle(hThread[0]) == FALSE)
                        {
                                printf("Error, failed in closing customer
thread handle with errno: %d\n", GetLastError());
                        }

                        if (CloseHandle(hThread[1]) == FALSE)
                        {
                                printf("Error, failed in closing history
thread handle with errno: %d\n", GetLastError());
                        }


                }

        }

        // flush the bulk connection
        rcint = bcp_done(c_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(c_hdbc1);

        rcint = bcp_done(c_hdbc2);
        if (rcint < 0)
                HandleErrorDBC(c_hdbc2);

        printf("Finished loading customer table.\n");

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxcuscl");

        // build non-clustered index
        if (aptr->build_index == 1)
                BuildIndex("idxcusnc");
```

```
        // Output the NURAND used for the loader into C_FIRST for C_ID =
1,
        // C_W_ID = 1, and C_D_ID = 1
        sprintf(cmd, "isql -S%s -U%s -P%s -d%s -e -Q\"update customer set
c_first = 'C_LOAD = %d' where c_id = 1 and c_w_id = 1 and c_d_id = 1\" >
logs\\nurand_load.log",
                                aptr->server,
                                aptr->user,
                                aptr->password,
                                aptr->database,
                                LOADER_NURAND_C);

        system(cmd);

        SQLFreeStmt(c_hstmt1, SQL_DROP);
        SQLDisconnect(c_hdbc1);
        SQLFreeConnect(c_hdbc1);

        SQLFreeStmt(c_hstmt2, SQL_DROP);
        SQLDisconnect(c_hdbc2);
        SQLFreeConnect(c_hdbc2);


    return;

}


//============================================================================
//
// Function   : CustomerBufInit
//
//============================================================================

void CustomerBufInit()
{
        int     i;

        for (i=0;i<customers_per_district;i++)
        {
                customer_buf[i].c_id = 0;
                customer_buf[i].c_d_id = 0;
                customer_buf[i].c_w_id = 0;

                strcpy(customer_buf[i].c_first,"");
                strcpy(customer_buf[i].c_middle,"");
                strcpy(customer_buf[i].c_last,"");
                strcpy(customer_buf[i].c_street_1,"");
                strcpy(customer_buf[i].c_street_2,"");
                strcpy(customer_buf[i].c_city,"");
                strcpy(customer_buf[i].c_state,"");
                strcpy(customer_buf[i].c_zip,"");
                strcpy(customer_buf[i].c_phone,"");
                strcpy(customer_buf[i].c_credit,"");

                customer_buf[i].c_credit_lim = 0;
                customer_buf[i].c_discount = (float) 0;

                //  fix to avoid ODBC float to numeric conversion problem.
                //      customer_buf[i].c_balance = 0;
                strcpy(customer_buf[i].c_balance,"");
```

```
                customer_buf[i].c_ytd_payment = 0;
                customer_buf[i].c_payment_cnt = 0;
                customer_buf[i].c_delivery_cnt = 0;

                strcpy(customer_buf[i].c_data,"");

                customer_buf[i].h_amount = 0;

                strcpy(customer_buf[i].h_data,"");

        }


}


//===============================================================================
//
// Function   : CustomerBufLoad
//
// Fills shared buffer for HISTORY and CUSTOMER
//===============================================================================

void CustomerBufLoad(int d_id, int w_id)
{
        long                                    i;
    CUSTOMER_SORT_STRUCT    c[CUSTOMERS_PER_DISTRICT];


        for (i=0;i<customers_per_district;i++)
        {
                if (i < 1000)
                        LastName(i, c[i].c_last);
                else
                        LastName(NURand(255,0,999,LOADER_NURAND_C),
c[i].c_last);

                MakeAlphaString(8,16,FIRST_NAME_LEN, c[i].c_first);

                c[i].c_id = i+1;

        }

        printf("...Loading customer buffer for: d_id = %d, w_id = %d\n",
                        d_id, w_id);

        for (i=0;i<customers_per_district;i++)
        {

                customer_buf[i].c_d_id = d_id;
                customer_buf[i].c_w_id = w_id;
                customer_buf[i].h_amount = 10.0;

                customer_buf[i].c_ytd_payment  = 10.0;

                customer_buf[i].c_payment_cnt  = 1;
                customer_buf[i].c_delivery_cnt = 0;

                // Generate CUSTOMER and HISTORY data

                customer_buf[i].c_id = c[i].c_id;

                strcpy(customer_buf[i].c_first, c[i].c_first);
                strcpy(customer_buf[i].c_last, c[i].c_last);

                customer_buf[i].c_middle[0] = 'O';
                customer_buf[i].c_middle[1] = 'E';

                MakeAddress(customer_buf[i].c_street_1,
                                customer_buf[i].c_street_2,
                                customer_buf[i].c_city,
                                customer_buf[i].c_state,
                                customer_buf[i].c_zip);

                MakeNumberString(16, 16, PHONE_LEN,
customer_buf[i].c_phone);

                if (RandomNumber(1L, 100L) > 10)
                        customer_buf[i].c_credit[0] = 'G';
                else
                        customer_buf[i].c_credit[0] = 'B';
                customer_buf[i].c_credit[1] = 'C';

                customer_buf[i].c_credit_lim = 50000.0;
                customer_buf[i].c_discount = ((float) RandomNumber(0L,
5000L)) / 10000.0;

                //  fix to avoid ODBC float to numeric conversion problem.

                //  customer_buf[i].c_balance = -10.0;
                strcpy(customer_buf[i].c_balance,"-10.0");

                MakeAlphaString(300, 500, C_DATA_LEN,
customer_buf[i].c_data);

                // Generate HISTORY data
                MakeAlphaString(12, 24, H_DATA_LEN,
customer_buf[i].h_data);

        }
}

//===============================================================================
//
// Function   : LoadCustomerTable
//
//===============================================================================

void LoadCustomerTable(LOADER_TIME_STRUCT *customer_time_start)
{
        int         i;
    long        c_id;
    short       c_d_id;
    short       c_w_id;
    char        c_first[FIRST_NAME_LEN+1];
    char        c_middle[MIDDLE_NAME_LEN+1];
    char        c_last[LAST_NAME_LEN+1];
    char        c_street_1[ADDRESS_LEN+1];
    char        c_street_2[ADDRESS_LEN+1];
    char        c_city[ADDRESS_LEN+1];
    char        c_state[STATE_LEN+1];
```

```
    char        c_zip[ZIP_LEN+1];
    char        c_phone[PHONE_LEN+1];
    char        c_credit[CREDIT_LEN+1];
    double      c_credit_lim;
    double      c_discount;

        //  fix to avoid ODBC float to numeric conversion problem.

        //  double            c_balance;
        char           c_balance[6];

    double      c_ytd_payment;
    short       c_payment_cnt;
    short       c_delivery_cnt;
    char        c_data[C_DATA_LEN+1];
        char           c_since[C_SINCE_LEN+1];
        RETCODE        rc;

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0, FIRST_NAME_LEN, NULL, 0,
0, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0, MIDDLE_NAME_LEN,NULL, 0,
0, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN, NULL, 0, 0,
6);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN, NULL, 0,
0, 7);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN, NULL,0,0,
8);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL, 0, 0,
9);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0, 0,
10);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0, 11);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL, 0, 0,
12);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0, C_SINCE_LEN, NULL, 0,
SQLCHARACTER, 13);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL, 0, 0,
14);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 15);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0, SQL_VARLEN_DATA, NULL,
0, SQLFLT8, 16);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);


        //  fix to avoid ODBC float to numeric conversion problem.

    //  rc = bcp_bind(c_hdbc1, (BYTE *) &c_balance, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 17);
        //  if (rc != SUCCEED)
        //      HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0,
SQLCHARACTER, 17);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);


    rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 18);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 19);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt,0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 20);
```

```
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0, 500, NULL, 0, 0, 21);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    for (i = 0; i < customers_per_district; i++)
    {
                c_id = customer_buf[i].c_id;
                c_d_id = customer_buf[i].c_d_id;
                c_w_id = customer_buf[i].c_w_id;

                strcpy(c_first, customer_buf[i].c_first);
                strcpy(c_middle, customer_buf[i].c_middle);
                strcpy(c_last, customer_buf[i].c_last);
                strcpy(c_street_1, customer_buf[i].c_street_1);
                strcpy(c_street_2, customer_buf[i].c_street_2);
                strcpy(c_city, customer_buf[i].c_city);
                strcpy(c_state, customer_buf[i].c_state);
                strcpy(c_zip, customer_buf[i].c_zip);
                strcpy(c_phone, customer_buf[i].c_phone);
                strcpy(c_credit, customer_buf[i].c_credit);

                FormatDate(&c_since);

                c_credit_lim = customer_buf[i].c_credit_lim;
                c_discount = customer_buf[i].c_discount;

                //  fix to avoid ODBC float to numeric conversion problem.

                //  c_balance = customer_buf[i].c_balance;
                strcpy(c_balance, customer_buf[i].c_balance);

                c_ytd_payment = customer_buf[i].c_ytd_payment;
                c_payment_cnt = customer_buf[i].c_payment_cnt;
                c_delivery_cnt = customer_buf[i].c_delivery_cnt;

                strcpy(c_data, customer_buf[i].c_data);

                // Send data to server
                rc = bcp_sendrow(c_hdbc1);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

                customer_rows_loaded++;
                CheckForCommit(c_hdbc1, c_hstmt1, customer_rows_loaded,
"customer", &customer_time_start->time_start);
    }

}


//===========================================================================
//
// Function   : LoadHistoryTable
//
//===========================================================================

void LoadHistoryTable(LOADER_TIME_STRUCT *history_time_start)
{
        int         i;
    long        c_id;
    short       c_d_id;
    short       c_w_id;
        double          h_amount;
    char        h_data[H_DATA_LEN+1];
        char            h_date[H_DATE_LEN+1];
        RETCODE         rc;

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

        rc = bcp_bind(c_hdbc2, (BYTE *) &h_date, 0, H_DATE_LEN, NULL, 0,
SQLCHARACTER, 6);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0, SQL_VARLEN_DATA, NULL,
0, SQLFLT8, 7);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0, H_DATA_LEN, NULL, 0, 0, 8);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

    for (i = 0; i < customers_per_district; i++)
    {
                c_id = customer_buf[i].c_id;
                c_d_id = customer_buf[i].c_d_id;
                c_w_id = customer_buf[i].c_w_id;
                h_amount = customer_buf[i].h_amount;
                strcpy(h_data, customer_buf[i].h_data);

                FormatDate(&h_date);

                // send to server
                rc = bcp_sendrow(c_hdbc2);
                if (rc != SUCCEED)
```

```
                        HandleErrorDBC(c_hdbc2);

                history_rows_loaded++;
                CheckForCommit(c_hdbc2, c_hstmt2, history_rows_loaded,
"history", &history_time_start->time_start);
        }

}


//============================================================-
===============
//
// Function   : LoadOrders
//
//============================================================================
====

void LoadOrders()
{
        LOADER_TIME_STRUCT      orders_time_start;
        LOADER_TIME_STRUCT      new_order_time_start;
        LOADER_TIME_STRUCT      order_line_time_start;
        short                           w_id;
    short                       d_id;
        DWORD                           dwThreadID[MAX_ORDER_THREADS];
        HANDLE                          hThread[MAX_ORDER_THREADS];
        char                            name[20];
        RETCODE                                 rc;
        char                                    bcphint[128];

        // seed with unique number
        seed(6);

        printf("Loading orders...\n");

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                        BuildIndex("idxordcl");
                        BuildIndex("idxnodcl");
                        BuildIndex("idxodlcl");
        }

        // initialize bulk copy
        sprintf(name, "%s..%s", aptr->database, "orders");

        rc = bcp_init(o_hdbc1, name, NULL, "logs\\orders.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (o_w_id, o_d_id, o_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
                rc = bcp_control(o_hdbc1, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);
        }

        sprintf(name, "%s..%s", aptr->database, "new_order");

        rc = bcp_init(o_hdbc2, name, NULL, "logs\\neword.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (no_w_id, no_d_id,
no_o_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 9000));
                rc = bcp_control(o_hdbc2, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc2);
        }

        sprintf(name, "%s..%s", aptr->database, "order_line");

        rc = bcp_init(o_hdbc3, name, NULL, "logs\\ordline.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (ol_w_id, ol_d_id,
ol_o_id, ol_number), ROWS_PER_BATCH = %u", (aptr->num_warehouses *
300000));
                rc = bcp_control(o_hdbc3, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
        }

        orders_rows_loaded      = 0;
        new_order_rows_loaded   = 0;
        order_line_rows_loaded  = 0;

        OrdersBufInit();

        orders_time_start.time_start = (TimeNow() / MILLI);
        new_order_time_start.time_start = (TimeNow() / MILLI);
        order_line_time_start.time_start = (TimeNow() / MILLI);

        for (w_id = (short)aptr->starting_warehouse; w_id <= aptr-
>num_warehouses; w_id++)
        {
                for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
                {

                        OrdersBufLoad(d_id, w_id);

                        // start parallel loading threads here...

                        // start Orders table thread

                        printf("...Loading Order Table for: d_id = %d, w_id
= %d\n", d_id, w_id);

                        hThread[0] = CreateThread(NULL,
                                                                                0,

(LPTHREAD_START_ROUTINE) LoadOrdersTable,

&orders_time_start,
```

```
                                                                0,
&dwThreadID[0]);

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

                        // start NewOrder table thread

                        printf("...Loading New-Order Table for: d_id = %d,
w_id = %d\n", d_id, w_id);

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

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

                        // start Order-Line table thread

                        printf("...Loading Order-Line Table for: d_id = %d,
w_id = %d\n", d_id, w_id);

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

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

                        WaitForSingleObject( hThread[0], INFINITE );
                        WaitForSingleObject( hThread[1], INFINITE );
                        WaitForSingleObject( hThread[2], INFINITE );

                        if (CloseHandle(hThread[0]) == FALSE)
                        {
                                printf("Error, failed in closing Orders
thread handle with errno: %d\n", GetLastError());
                        }

                        if (CloseHandle(hThread[1]) == FALSE)
                        {
                                printf("Error, failed in closing NewOrder
thread handle with errno: %d\n", GetLastError());
                        }

                        if (CloseHandle(hThread[2]) == FALSE)
                        {
                                printf("Error, failed in closing OrderLine
thread handle with errno: %d\n", GetLastError());
                        }

                }
        }

        printf("Finished loading orders.\n");


    return;
}


//==============================================================================
//
// Function   : OrdersBufInit
//
// Clears shared buffer for ORDERS, NEWORDER, and ORDERLINE
//
//==============================================================================

void OrdersBufInit()
{
        int     i;
        int             j;

        for (i=0;i<orders_per_district;i++)
        {
                orders_buf[i].o_id = 0;
                orders_buf[i].o_d_id = 0;
                orders_buf[i].o_w_id = 0;
                orders_buf[i].o_c_id = 0;
                orders_buf[i].o_carrier_id = 0;
                orders_buf[i].o_ol_cnt = 0;
                orders_buf[i].o_all_local = 0;

                for (j=0;j<=14;j++)
                {
                        orders_buf[i].o_ol[j].ol = 0;
                        orders_buf[i].o_ol[j].ol_i_id = 0;
                        orders_buf[i].o_ol[j].ol_supply_w_id = 0;
                        orders_buf[i].o_ol[j].ol_quantity = 0;
                        orders_buf[i].o_ol[j].ol_amount = 0;
                        strcpy(orders_buf[i].o_ol[j].ol_dist_info,"");
                }
        }
```

```
}


//============================================================================
//
// Function   : OrdersBufLoad
//
// Fills shared buffer for ORDERS, NEWORDER, and ORDERLINE
//
//============================================================================

void OrdersBufLoad(int d_id, int w_id)
{

        int     cust[ORDERS_PER_DISTRICT+1];
        long    o_id;
        short   ol;

        printf("...Loading Order Buffer for: d_id = %d, w_id = %d\n",
                        d_id, w_id);

        GetPermutation(cust, orders_per_district);

        for (o_id=0;o_id<orders_per_district;o_id++)
        {

                // Generate ORDER and NEW-ORDER data

                orders_buf[o_id].o_d_id = d_id;
                orders_buf[o_id].o_w_id = w_id;
                orders_buf[o_id].o_id = o_id+1;
                orders_buf[o_id].o_c_id = cust[o_id+1];
                orders_buf[o_id].o_ol_cnt = (short)RandomNumber(5L, 15L);

                if (o_id < first_new_order)
                {
                        orders_buf[o_id].o_carrier_id =
(short)RandomNumber(1L, 10L);
                        orders_buf[o_id].o_all_local  = 1;
                }
                else
                {
                        orders_buf[o_id].o_carrier_id = 0;
                        orders_buf[o_id].o_all_local  = 1;
                }

                for (ol=0; ol<orders_buf[o_id].o_ol_cnt; ol++)
                {

                        orders_buf[o_id].o_ol[ol].ol = ol+1;
                        orders_buf[o_id].o_ol[ol].ol_i_id =
RandomNumber(1L, max_items);
                        orders_buf[o_id].o_ol[ol].ol_supply_w_id = w_id;
                        orders_buf[o_id].o_ol[ol].ol_quantity = 5;
                        MakeAlphaString(24, 24, OL_DIST_INFO_LEN,
&orders_buf[o_id].o_ol[ol].ol_dist_info);

                        // Generate ORDER-LINE data
                        if (o_id < first_new_order)
                        {
                                orders_buf[o_id].o_ol[ol].ol_amount = 0;
                                // Added to insure ol_delivery_d set
properly during load

        FormatDate(&orders_buf[o_id].o_ol[ol].ol_delivery_d);

                        }
                        else
                        {
                                orders_buf[o_id].o_ol[ol].ol_amount =
RandomNumber(1,999999)/100.0;
                                // Added to insure ol_delivery_d set
properly during load

                                // odbc datetime format

        strcpy(orders_buf[o_id].o_ol[ol].ol_delivery_d,"1899-12-31
00:00:00.000");

                        }
                }
        }
}


//============================================================================
//
// Function   : LoadOrdersTable
//
//============================================================================

void LoadOrdersTable(LOADER_TIME_STRUCT *orders_time_start)
{
        int         i;
    long        o_id;
        short           o_d_id;
        short           o_w_id;
    long        o_c_id;
    short       o_carrier_id;
    short       o_ol_cnt;
    short       o_all_local;
        char            o_entry_d[O_ENTRY_D_LEN+1];
        RETCODE         rc;
        DBINT           rcint;

        // bind ORDER data
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);
```

```
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

        rc = bcp_bind(o_hdbc1, (BYTE *) &o_entry_d, 0, O_ENTRY_D_LEN,
NULL, 0, SQLCHARACTER, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 6);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 7);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 8);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    for (i = 0; i < orders_per_district; i++)
    {
                o_id         = orders_buf[i].o_id;
                o_d_id       = orders_buf[i].o_d_id;
                o_w_id       = orders_buf[i].o_w_id;
                o_c_id       = orders_buf[i].o_c_id;
                o_carrier_id = orders_buf[i].o_carrier_id;
                o_ol_cnt     = orders_buf[i].o_ol_cnt;
                o_all_local  = orders_buf[i].o_all_local;

                FormatDate(&o_entry_d);

                // send data to server
                rc = bcp_sendrow(o_hdbc1);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);

                orders_rows_loaded++;
                CheckForCommit(o_hdbc1, o_hstmt1, orders_rows_loaded,
"orders", &orders_time_start->time_start);
    }

        // rcint = bcp_batch(o_hdbc1);
        // if (rcint < 0)
        //      HandleErrorDBC(o_hdbc1);

        if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
        {
                rcint = bcp_done(o_hdbc1);
                if (rcint < 0)
                        HandleErrorDBC(o_hdbc1);

                SQLFreeStmt(o_hstmt1, SQL_DROP);
                SQLDisconnect(o_hdbc1);
                SQLFreeConnect(o_hdbc1);

                // if build index after load...
                if ((aptr->build_index == 1) && (aptr->index_order == 0))
                        BuildIndex("idxordcl");

                // build non-clustered index
                if (aptr->build_index == 1)
                        BuildIndex("idxordnc");
        }

}


//============================================================================
//
// Function   : LoadNewOrderTable
//
//============================================================================

void LoadNewOrderTable(LOADER_TIME_STRUCT *new_order_time_start)
{
    int         i;
    long        o_id;
    short       o_d_id;
    short       o_w_id;
        RETCODE         rc;
        DBINT           rcint;

        // Bind NEW-ORDER data

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

        for (i = first_new_order; i < last_new_order; i++)
        {
                o_id    = orders_buf[i].o_id;
                o_d_id  = orders_buf[i].o_d_id;
                o_w_id  = orders_buf[i].o_w_id;

                rc = bcp_sendrow(o_hdbc2);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc2);

                new_order_rows_loaded++;
                CheckForCommit(o_hdbc2, o_hstmt2, new_order_rows_loaded,
"new_order", &new_order_time_start->time_start);
        }

        // rcint = bcp_batch(o_hdbc2);
        // if (rcint < 0)
```

```
        //      HandleErrorDBC(o_hdbc2);

        if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
        {
                rcint = bcp_done(o_hdbc2);
                if (rcint < 0)
                        HandleErrorDBC(o_hdbc2);

                SQLFreeStmt(o_hstmt2, SQL_DROP);
                SQLDisconnect(o_hdbc2);
                SQLFreeConnect(o_hdbc2);

                // if build index after load...
                if ((aptr->build_index == 1) && (aptr->index_order == 0))
                        BuildIndex("idxnodcl");

        }

}


//============================================================================
//
// Function   : LoadOrderLineTable
//
//============================================================================

void LoadOrderLineTable(LOADER_TIME_STRUCT *order_line_time_start)
{
        int         i,j;
    long        o_id;
        short           o_d_id;
        short           o_w_id;
    long        ol;
        long            ol_i_id;
    short       ol_supply_w_id;
    short       ol_quantity;
    double      ol_amount;
    char        ol_dist_info[DIST_INFO_LEN+1];
        char            ol_delivery_d[OL_DELIVERY_D_LEN+1];
        RETCODE         rc;
        DBINT           rcint;

        // bind ORDER-LINE data
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 6);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

        rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0,
OL_DELIVERY_D_LEN, NULL, 0, SQLCHARACTER, 7);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 8);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0, SQL_VARLEN_DATA, NULL,
0, SQLFLT8, 9);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0, DIST_INFO_LEN, NULL,
0, 0, 10);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

        for (i = 0; i < orders_per_district; i++)
    {
                o_id    = orders_buf[i].o_id;
                o_d_id  = orders_buf[i].o_d_id;
                o_w_id  = orders_buf[i].o_w_id;

                for (j=0; j < orders_buf[i].o_ol_cnt; j++)
                {
                        ol              = orders_buf[i].o_ol[j].ol;
                        ol_i_id         = orders_buf[i].o_ol[j].ol_i_id;
                        ol_supply_w_id =
orders_buf[i].o_ol[j].ol_supply_w_id;
                        ol_quantity     = orders_buf[i].o_ol[j].ol_quantity;
                        ol_amount       = orders_buf[i].o_ol[j].ol_amount;


        strcpy(ol_delivery_d,orders_buf[i].o_ol[j].ol_delivery_d);


        strcpy(ol_dist_info,orders_buf[i].o_ol[j].ol_dist_info);

                        rc = bcp_sendrow(o_hdbc3);
                        if (rc != SUCCEED)
                                HandleErrorDBC(o_hdbc3);

                        order_line_rows_loaded++;
                        CheckForCommit(o_hdbc3, o_hstmt3,
order_line_rows_loaded, "order_line", &order_line_time_start->time_start);
```

```
                }

        }

        // rcint = bcp_batch(o_hdbc3);
        // if (rcint < 0)
        //      HandleErrorDBC(o_hdbc3);

        if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
        {
                rcint = bcp_done(o_hdbc3);
                if (rcint < 0)
                        HandleErrorDBC(o_hdbc3);

                SQLFreeStmt(o_hstmt3, SQL_DROP);
                SQLDisconnect(o_hdbc3);
                SQLFreeConnect(o_hdbc3);

                // if build index after load...
                if ((aptr->build_index == 1) && (aptr->index_order == 0))
                        BuildIndex("idxodlcl");

        }

}


//==============================================================================
//
// Function   : GetPermutation
//
//==============================================================================

void GetPermutation(int perm[], int n)
{
        int i, r, t;

        for (i=1;i<=n;i++)
                perm[i] = i;

        for (i=1;i<=n;i++)
        {
                r = RandomNumber(i,n);
                t = perm[i];
                perm[i] = perm[r];
                perm[r] = t;
        }

}


//==============================================================================
//
// Function   : CheckForCommit
//
//==============================================================================

void CheckForCommit(HDBC hdbc,
                                HSTMT hstmt,
                        int rows_loaded,
                                char *table_name,
                                long *time_start)
{

    long       time_end, time_diff;
       // DBINT       rcint;

    if ( !(rows_loaded % aptr->batch) )
    {

                // rcint = bcp_batch(hdbc);
                // if (rcint < 0)
                //      HandleErrorDBC(hdbc);

                time_end = (TimeNow() / MILLI);
                time_diff = time_end - *time_start;

                printf("-> Loaded %ld rows into %s in %ld sec - Total = %d
(%.2f rps)\n",
                        aptr->batch,
                        table_name,
                                time_diff,
                        rows_loaded,
                                (float) aptr->batch / (time_diff ? time_diff
: 1L));

                *time_start = time_end;
    }

    return;
}


//==============================================================================
//
// Function   : OpenConnections
//
//==============================================================================

void OpenConnections()
{

        RETCODE         rc;

        char                    szDriverString[300];
        char                    szDriverStringOut[1024];
        SQLSMALLINT             cbDriverStringOut;


        SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv );

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

        SQLAllocHandle(SQL_HANDLE_DBC, henv , &i_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &w_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc2);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc2);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc3);
```

```
        SQLSetConnectAttr(i_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
        SQLSetConnectAttr(w_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
        SQLSetConnectAttr(c_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
        SQLSetConnectAttr(c_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
        SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
        SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
        SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );

        // Open connections to SQL Server

        // Connection 1

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


        rc = SQLSetConnectOption (i_hdbc1, SQL_PACKET_SIZE, aptr-
>pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        rc = SQLDriverConnect ( i_hdbc1,
                                        NULL,
                                            (SQLCHAR*)&szDriverString[0]
,
                                            SQL_NTS,

        (SQLCHAR*)&szDriverStringOut[0],
                                            sizeof(szDriverStringOut),
                                            &cbDriverStringOut,
                                            SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        // Connection 2

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


        rc = SQLSetConnectOption (w_hdbc1, SQL_PACKET_SIZE, aptr-
>pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = SQLDriverConnect ( w_hdbc1,
                                                        NULL,

        (SQLCHAR*)&szDriverString[0] ,
                                                        SQL_NTS,

        (SQLCHAR*)&szDriverStringOut[0],

        sizeof(szDriverStringOut),
                                                        &cbDriverStringOut,
                                                        SQL_DRIVER_NOPROMPT
);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        // Connection 3

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

        rc = SQLSetConnectOption (c_hdbc1, SQL_PACKET_SIZE, aptr-
>pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

        rc = SQLDriverConnect ( c_hdbc1,
                                                        NULL,

        (SQLCHAR*)&szDriverString[0] ,
                                                        SQL_NTS,

        (SQLCHAR*)&szDriverStringOut[0],

        sizeof(szDriverStringOut),
                                                        &cbDriverStringOut,
                                                        SQL_DRIVER_NOPROMPT
);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

        // Connection 4

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


        rc = SQLSetConnectOption (c_hdbc2, SQL_PACKET_SIZE, aptr-
>pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

        rc = SQLDriverConnect ( c_hdbc2,
                                                        NULL,

        (SQLCHAR*)&szDriverString[0] ,
                                                        SQL_NTS,
```

```
        (SQLCHAR*)&szDriverStringOut[0],

        sizeof(szDriverStringOut),
                                                &cbDriverStringOut,
                                                SQL_DRIVER_NOPROMPT
);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

        // Connection 5

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

        rc = SQLSetConnectOption (o_hdbc1, SQL_PACKET_SIZE, aptr-
>pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

        rc = SQLDriverConnect ( o_hdbc1,
                                                NULL,

        (SQLCHAR*)&szDriverString[0] ,
                                                SQL_NTS,

        (SQLCHAR*)&szDriverStringOut[0],

        sizeof(szDriverStringOut),
                                                &cbDriverStringOut,
                                                SQL_DRIVER_NOPROMPT
);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

        // Connection 6

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

        rc = SQLSetConnectOption (o_hdbc2, SQL_PACKET_SIZE, aptr-
>pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

        rc = SQLDriverConnect ( o_hdbc2,
                                                NULL,

        (SQLCHAR*)&szDriverString[0] ,
                                                SQL_NTS,

        (SQLCHAR*)&szDriverStringOut[0],

        sizeof(szDriverStringOut),
                                                &cbDriverStringOut,
                                                SQL_DRIVER_NOPROMPT
);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

        // Connection 7

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

        rc = SQLSetConnectOption (o_hdbc3, SQL_PACKET_SIZE, aptr-
>pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

        rc = SQLDriverConnect ( o_hdbc3,
                                                NULL,

        (SQLCHAR*)&szDriverString[0] ,
                                                SQL_NTS,

        (SQLCHAR*)&szDriverStringOut[0],

        sizeof(szDriverStringOut),
                                                &cbDriverStringOut,
                                                SQL_DRIVER_NOPROMPT
);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

}

//============================================================================
//
// Function name: BuildIndex
//
//============================================================================

void BuildIndex(char   *index_script)
{
        char    cmd[256];

        printf("Starting index creation:  %s\n",index_script);

        sprintf(cmd, "isql -S%s -U%s -P%s -e -i%s\\%s.sql > logs\\%s.log",
                                aptr->server,
                                aptr->user,
                                aptr->password,
                                aptr->index_script_path,
                                index_script,
                                index_script);

        system(cmd);
```

```
        printf("Finished index creation:  %s\n",index_script);
}


void HandleErrorDBC (SQLHDBC  hdbc1)
{

        SQLCHAR        SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
        SQLINTEGER  NativeError;
        SQLSMALLINT i, MsgLen;
        SQLRETURN   rc2;
        char           timebuf[128];
        char           datebuf[128];
        FILE           *fp1;

        i = 1;
        while (( rc2 = SQLGetDiagRec(SQL_HANDLE_DBC , hdbc1, i, SqlState ,
&NativeError,
                                Msg, sizeof(Msg) , &MsgLen )) !=
SQL_NO_DATA )
        {

                sprintf( szLastError , "%s" ,  Msg );

                _strtime(timebuf);
                _strdate(datebuf);

                printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

                fp1 = fopen("logs\\tpccldr.err","w");
                if (fp1 == NULL)
                        printf("ERROR:  Unable to open errorlog file.\n");
                else
                {
                        fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);
                        fclose(fp1);
                }

                i++;

        }
}

void HandleErrorSTMT (HSTMT  hstmt1)
{

        SQLCHAR        SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
        SQLINTEGER  NativeError;
        SQLSMALLINT i, MsgLen;
        SQLRETURN   rc2;
        char           timebuf[128];
        char           datebuf[128];
        FILE           *fp1;

        i = 1;
        while (( rc2 = SQLGetDiagRec(SQL_HANDLE_STMT , hstmt1, i, SqlState
, &NativeError,
                                Msg, sizeof(Msg) , &MsgLen )) !=
SQL_NO_DATA )
        {

                sprintf( szLastError , "%s" ,  Msg );

                _strtime(timebuf);
                _strdate(datebuf);

                printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

                fp1 = fopen("logs\\tpccldr.err","w");
                if (fp1 == NULL)
                        printf("ERROR:  Unable to open errorlog file.\n");
                else
                {
                        fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);
                        fclose(fp1);
                }

                i++;

        }
}

void FormatDate ( char* szTimeCOutput )
{

        struct tm when;
        time_t now;

        time( &now );
        when  = *localtime( &now );

        mktime( &when );

        // odbc datetime format
        strftime( szTimeCOutput , 30 , "%Y-%m-%d %H:%M:%S.000", &when );

        return;

}


//============================================================================
//
// Function   : CheckSQL
//
//============================================================================

void CheckSQL()
{

        RETCODE         rc;

        char                    szDriverString[300];
        char                    szDriverStringOut[1024];
        int                             SQLBuildFlag;

        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 7.00.623 or higher

        SQLBuildFlag = 1;

        if ( SQLVersion[0] == 55 )
        {
                if ( SQLVersion[2] == 48 )
                {
                        if ( SQLVersion[5] == 56 )
                        {
                                if ( (SQLVersion[6] >= 48) &  (SQLVersion[7]
>= 53) )
                                {
                                        SQLBuildFlag = 0;
                                        printf("You are using SQL Server
version = %9s\n\n", SQLVersion);
                                }
                                else
                                {
                                        SQLBuildFlag = 1;
                                }
                        }
                        else
                        {
                                if ( SQLVersion[5] >= 54 )
                                {
                                        if ( (SQLVersion[6] >= 50) &
(SQLVersion[7] >= 51) )
                                        {
                                                SQLBuildFlag = 0;
                                                printf("You are using SQL
Server version = %9s\n\n", SQLVersion);
                                        }
                                        else
                                        {
                                                SQLBuildFlag = 1;
                                        }
                                }
                                else
                                {
                                        if ( SQLVersion[5] >= 55 )
                                        {
                                                if ( (SQLVersion[6] >= 48) &
(SQLVersion[7] >= 48) )
                                                {
                                                        SQLBuildFlag = 0;
                                                        printf("You are using
SQL Server version = %9s\n\n", SQLVersion);
                                                }
                                                else
                                                {
                                                        SQLBuildFlag = 1;
                                                }
                                        }
                                }

                        }
                }
                else
                {
                        if ( SQLVersion[5] >= 49 )
                        {
```

```
                                if ( (SQLVersion[6] >= 52) & (SQLVersion[7]
>= 48) )
                                {
                                        SQLBuildFlag = 0;
                                        printf("You are using SQL Server
version = %9s\n\n", SQLVersion);
                                }
                                else
                                {
                                        SQLBuildFlag = 1;
                                }
                        }
                        else
                        {
                                SQLBuildFlag = 1;
                        }
                }
        }
        else
        {
                SQLBuildFlag = 1;
        }


        if ( SQLBuildFlag == 1 )
        {
                printf("ERROR.  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
7.00.623 or better.\n");
                printf("and re-run the SETUP program.\n\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;
}
```

Appendix C - Tunable Parameters

Microsoft SQL Server Startup Parameters

C:\MSSQL\BINN\SQLSERVR.EXE -c -x -t3502 -g37

Where:

- -c Start SQL Server independently of the Service Control Manager
- -x Disables the keeping of CPU time and cache hit ratio statistics
- -t3502 Writes a message to the SQL Server Errorlog showing the beginning and ending time of each checkpoint
- -g37 Specifies the amount of virtual address space, in MB, SQL Server will leave available for memory allocations, excluding the buffer pool and thread stacks, such as dynamically-loaded DLLs, extended procedure calls, etc.. If this option is not specified, SQL Server will use a value that is suitable for a wide range of runtime environments. Use of this option may be appropriate in 2GB (3GB Enterprise Edition) configurations in which the memory usage requirements of SQL Server are atypical and the virtual address space of the SQL Server process is totally in use. Incorrect use of this option can lead to conditions under which SQL Server may not start or may encounter runtime errors.

SQL Server Stack Size

The default stack size for Microsoft SQL Server 7.0 was changed using the EDITBIN utility. The EDITBIN utility ships with Microsoft Visual C++ V6.0. The command used to change the stack size is:

editbin /S: 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.

BOOT.INI

The /3gb switch was added to the boot.ini file to cause Windows NT Enterprise Edition to allow 3GB of user and 1GB of kernel virtual address space, rather than the usual 2GB of virtual address space for each.

Microsoft SQL Server Configuration Parameters

```
1> 2> 3> 4> 5> 6> 7> 8> 9> 10> 11>
--  File:     VERSION.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Returns SQL Server version string



print " "
select convert(char(30), getdate(),9)
print " "

 ------------------------------
 Oct 21 1999 12:44:10:107PM

(1 row affected)

1> 2> 3>
select @@version




 --------------------------------------------------------------------------
-----
       ------------------------------------------------------------------
-----
       ------------------------------------------------------------------
-----
       ---------------------------------
 Microsoft SQL Server  7.00 - 7.00.805 (Intel X86)
       Jun 11 1999 11:48:12
       Cop
       yright (c) 1988-1998 Microsoft Corporation
       Enterprise Edition on Windo
       ws NT 4.0 (Build 1381: Service Pack 4)
```

```
(1 row affected)
1> 2>
1> 2> 3> 4> 5> 6> 7> 8> 9> 10>
--  File:     CONFIG.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Collects SQL Server configuration parameters


print " "
select convert(char(30), getdate(),9)
print " "

 ------------------------------
 Oct 21 1999 12:44:11:937PM

(1 row affected)

1> 2> 3> DBCC execution completed. If DBCC printed error messages, contact
your system administrator.
Configuration option changed. Run the RECONFIGURE statement to install.

sp_configure "show advanced",1
1> 2> reconfigure with override
1> 2> sp_configure
 name
minimum     maximum     config_value run_value
 ------------------------------------------------------------------------ --
--------- ----------- ------------ -----------
 affinity mask
0  2147483647           15          15
 allow updates
0           1            0           0
 cost threshold for parallelism
0       32767            5           5
 cursor threshold
-1  2147483647           -1          -1
 default language
0        9999            0           0
 default sortorder id
0         255           50          50
 extended memory size (MB)
0  2147483647            0           0
 fill factor (%)
0         100            0           0
 index create memory (KB)
704     1600000            0           0
 language in cache
3         100            3           3
 language neutral full-text
0           1            0           0
 lightweight pooling
0           1            1           1
 locks
5000  2147483647            0           0
 max async IO
1         255          255         255
 max degree of parallelism
0          32            1           1
 max server memory (MB)
4  2147483647         3025        3025
 max text repl size (B)
0  2147483647        65536       65536
 max worker threads
10        1024          214         214
 media retention
0         365            0           0
 min memory per query (KB)
512  2147483647          512         512
 min server memory (MB)
0  2147483647         3025        3025
 nested triggers
0           1            0           0
 network packet size (B)
512       65535         4096        4096
 open objects
0  2147483647            0           0
 priority boost
0           1            1           1
 query governor cost limit
0  2147483647            0           0
 query wait (s)
-1  2147483647           -1          -1
 recovery interval (min)
0       32767        32767       32767
 remote access
0           1            0           0
 remote login timeout (s)
0  2147483647            5           5
 remote proc trans
0           1            0           0
 remote query timeout (s)
0  2147483647            0           0
 resource timeout (s)
5  2147483647           10          10
 scan for startup procs
0           1            0           0
 set working set size
0           1            1           1
 show advanced options
0           1            1           1
 spin counter
1  2147483647        10000       10000
 time slice (ms)
50        1000          100         100
 two digit year cutoff
1753        9999         2049        2049
 Unicode comparison style
0  2147483647            0           0
 Unicode locale id
0  2147483647        33280       33280
 user connections
0       32767            0           0
 user options
0        4095            0           0

1>
```

Internal RAID Configuration Parameters

```
Adapter No: 0

Number of Logical Drives : 1

Logical Drive = 0
        Capacity = 69982MB
        Span Depth = 1
        Raid Level = 1,
        Read Ahead = ADAPTIVE
        Stripe Size = 64KB,
        Status = OPTIMAL
        Write Policy = WRITE_THRU,
        Direct IO = DIRECT_IO,
        Number of Stripes = 2
        SPAN Number = 0
                Starting Block = 0
                Number of blocks = 143323136
                Device Number = 0
                        Channel Number = 0
                        Target Number = 15
                Device Number = 1
                        Channel Number = 1
                        Target Number = 15


 (Channel 0, ID 15)
        Type = HARDDISK,       Current Status = ONLINE
        Size 143323136 blocks, 69982MB  IFT  3101  0222

 (Channel 1, ID 15)
        Type = HARDDISK,       Current Status = ONLINE
        Size 143323136 blocks, 69982MB  IFT  3101  0222


Adapter No: 1

Number of Logical Drives : 1

Logical Drive = 0
        Capacity = 279904MB
        Span Depth = 4
        Raid Level = 0,
        Read Ahead = NORMAL
        Stripe Size = 64KB,
        Status = OPTIMAL
        Write Policy = WRITE_THRU,
        Direct IO = DIRECT_IO,
        Number of Stripes = 8
        SPAN Number = 0
                Starting Block = 0
                Number of blocks = 17913856
                Device Number = 0
                        Channel Number = 0
                        Target Number = 8
                Device Number = 1
                        Channel Number = 0
                        Target Number = 9
                Device Number = 2
                        Channel Number = 0
                        Target Number = 10
                Device Number = 3
                        Channel Number = 0
                        Target Number = 11
                Device Number = 4
                        Channel Number = 0
                        Target Number = 12
                Device Number = 5
                        Channel Number = 0
                        Target Number = 13
                Device Number = 6
                        Channel Number = 0
                        Target Number = 14
                Device Number = 7
                        Channel Number = 0
                        Target Number = 15
        SPAN Number = 1
                Starting Block = 0
                Number of blocks = 17913856
                Device Number = 0
                        Channel Number = 1
                        Target Number = 8
                Device Number = 1
                        Channel Number = 1
                        Target Number = 9
                Device Number = 2
                        Channel Number = 1
                        Target Number = 10
                Device Number = 3
                        Channel Number = 1
                        Target Number = 11
                Device Number = 4
                        Channel Number = 1
                        Target Number = 12
                Device Number = 5
                        Channel Number = 1
                        Target Number = 13
                Device Number = 6
                        Channel Number = 1
                        Target Number = 14
                Device Number = 7
                        Channel Number = 1
                        Target Number = 15
        SPAN Number = 2
                Starting Block = 0
                Number of blocks = 17913856
                Device Number = 0
                        Channel Number = 2
                        Target Number = 8
                Device Number = 1
                        Channel Number = 2
                        Target Number = 9
                Device Number = 2
                        Channel Number = 2
                        Target Number = 10
                Device Number = 3
```

```
                Channel Number = 2
                Target Number = 11
        Device Number = 4
                Channel Number = 2
                Target Number = 12
        Device Number = 5
                Channel Number = 2
                Target Number = 13
        Device Number = 6
                Channel Number = 2
                Target Number = 14
        Device Number = 7
                Channel Number = 2
                Target Number = 15
SPAN Number = 3
        Starting Block = 0
        Number of blocks = 17913856
        Device Number = 0
                Channel Number = 3
                Target Number = 8
        Device Number = 1
                Channel Number = 3
                Target Number = 9
        Device Number = 2
                Channel Number = 3
                Target Number = 10
        Device Number = 3
                Channel Number = 3
                Target Number = 11
        Device Number = 4
                Channel Number = 3
                Target Number = 12
        Device Number = 5
                Channel Number = 3
                Target Number = 13
        Device Number = 6
                Channel Number = 3
                Target Number = 14
        Device Number = 7
                Channel Number = 3
                Target Number = 15


(Channel 0, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603
```

```
 (Channel 2, ID 11)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 2, ID 12)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 2, ID 13)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 2, ID 14)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 2, ID 15)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 8)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 9)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 10)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 11)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 12)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 13)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 14)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 15)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603


Adapter No: 2

Number of Logical Drives : 1

Logical Drive = 0
       Capacity = 279904MB
       Span Depth = 4
       Raid Level = 0,
       Read Ahead = NORMAL
       Stripe Size = 64KB,
       Status = OPTIMAL
       Write Policy = WRITE_THRU,
       Direct IO = DIRECT_IO,
       Number of Stripes = 8
       SPAN Number = 0
              Starting Block = 0
              Number of blocks = 17913856
              Device Number = 0
                     Channel Number = 0
                     Target Number = 8
              Device Number = 1
                     Channel Number = 0
                     Target Number = 9
              Device Number = 2
                     Channel Number = 0
                     Target Number = 10
              Device Number = 3
                     Channel Number = 0
                     Target Number = 11
              Device Number = 4
                     Channel Number = 0
                     Target Number = 12
              Device Number = 5
                     Channel Number = 0
                     Target Number = 13
              Device Number = 6
                     Channel Number = 0
                     Target Number = 14
              Device Number = 7
                     Channel Number = 0
                     Target Number = 15
       SPAN Number = 1
              Starting Block = 0
              Number of blocks = 17913856
              Device Number = 0
                     Channel Number = 1
                     Target Number = 8
              Device Number = 1
                     Channel Number = 1
                     Target Number = 9
              Device Number = 2
                     Channel Number = 1
                     Target Number = 10
              Device Number = 3
                     Channel Number = 1
                     Target Number = 11
              Device Number = 4
                     Channel Number = 1
                     Target Number = 12
              Device Number = 5
                     Channel Number = 1
                     Target Number = 13
              Device Number = 6
                     Channel Number = 1
```

```
                Target Number = 14
        Device Number = 7
                Channel Number = 1
                Target Number = 15
SPAN Number = 2
        Starting Block = 0
        Number of blocks = 17913856
        Device Number = 0
                Channel Number = 2
                Target Number = 8
        Device Number = 1
                Channel Number = 2
                Target Number = 9
        Device Number = 2
                Channel Number = 2
                Target Number = 10
        Device Number = 3
                Channel Number = 2
                Target Number = 11
        Device Number = 4
                Channel Number = 2
                Target Number = 12
        Device Number = 5
                Channel Number = 2
                Target Number = 13
        Device Number = 6
                Channel Number = 2
                Target Number = 14
        Device Number = 7
                Channel Number = 2
                Target Number = 15
SPAN Number = 3
        Starting Block = 0
        Number of blocks = 17913856
        Device Number = 0
                Channel Number = 3
                Target Number = 8
        Device Number = 1
                Channel Number = 3
                Target Number = 9
        Device Number = 2
                Channel Number = 3
                Target Number = 10
        Device Number = 3
                Channel Number = 3
                Target Number = 11
        Device Number = 4
                Channel Number = 3
                Target Number = 12
        Device Number = 5
                Channel Number = 3
                Target Number = 13
        Device Number = 6
                Channel Number = 3
                Target Number = 14
        Device Number = 7
                Channel Number = 3
                Target Number = 15
```

```
(Channel 0, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
```

```
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 1, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 2, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 2, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 2, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 2, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 2, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 2, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 2, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 2, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603


Adapter No: 3

Number of Logical Drives : 1

Logical Drive = 0
      Capacity = 279904MB
      Span Depth = 4
      Raid Level = 0,
      Read Ahead = NORMAL
      Stripe Size = 64KB,
      Status = OPTIMAL
      Write Policy = WRITE_THRU,
      Direct IO = DIRECT_IO,
      Number of Stripes = 8
      SPAN Number = 0
            Starting Block = 0
            Number of blocks = 17913856
            Device Number = 0
                  Channel Number = 0
                  Target Number = 8
            Device Number = 1
                  Channel Number = 0
                  Target Number = 9
            Device Number = 2
                  Channel Number = 0
                  Target Number = 10
            Device Number = 3
                  Channel Number = 0
                  Target Number = 11
            Device Number = 4
                  Channel Number = 0
                  Target Number = 12
            Device Number = 5
                  Channel Number = 0
                  Target Number = 13
            Device Number = 6
                  Channel Number = 0
                  Target Number = 14
            Device Number = 7
                  Channel Number = 0
                  Target Number = 15
      SPAN Number = 1
            Starting Block = 0
            Number of blocks = 17913856
            Device Number = 0
                  Channel Number = 1
                  Target Number = 8
```

```
        Device Number = 1
                Channel Number = 1
                Target Number = 9
        Device Number = 2
                Channel Number = 1
                Target Number = 10
        Device Number = 3
                Channel Number = 1
                Target Number = 11
        Device Number = 4
                Channel Number = 1
                Target Number = 12
        Device Number = 5
                Channel Number = 1
                Target Number = 13
        Device Number = 6
                Channel Number = 1
                Target Number = 14
        Device Number = 7
                Channel Number = 1
                Target Number = 15
SPAN Number = 2
        Starting Block = 0
        Number of blocks = 17913856
        Device Number = 0
                Channel Number = 2
                Target Number = 8
        Device Number = 1
                Channel Number = 2
                Target Number = 9
        Device Number = 2
                Channel Number = 2
                Target Number = 10
        Device Number = 3
                Channel Number = 2
                Target Number = 11
        Device Number = 4
                Channel Number = 2
                Target Number = 12
        Device Number = 5
                Channel Number = 2
                Target Number = 13
        Device Number = 6
                Channel Number = 2
                Target Number = 14
        Device Number = 7
                Channel Number = 2
                Target Number = 15
SPAN Number = 3
        Starting Block = 0
        Number of blocks = 17913856
        Device Number = 0
                Channel Number = 3
                Target Number = 8
        Device Number = 1
                Channel Number = 3
                Target Number = 9
        Device Number = 2
                Channel Number = 3
                Target Number = 10
        Device Number = 3
                Channel Number = 3
                Target Number = 11
        Device Number = 4
                Channel Number = 3
                Target Number = 12
        Device Number = 5
                Channel Number = 3
                Target Number = 13
        Device Number = 6
                Channel Number = 3
                Target Number = 14
        Device Number = 7
                Channel Number = 3
                Target Number = 15


(Channel 0, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 10)
```

```
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603


Adapter No: 4

Number of Logical Drives : 1

Logical Drive = 0
       Capacity = 279904MB
       Span Depth = 4
       Raid Level = 0,
       Read Ahead = NORMAL
       Stripe Size = 64KB,
       Status = OPTIMAL
       Write Policy = WRITE_THRU,
       Direct IO = DIRECT_IO,
       Number of Stripes = 8
       SPAN Number = 0
              Starting Block = 0
              Number of blocks = 17913856
              Device Number = 0
                     Channel Number = 0
                     Target Number = 8
              Device Number = 1
                     Channel Number = 0
                     Target Number = 9
              Device Number = 2
                     Channel Number = 0
                     Target Number = 10
              Device Number = 3
                     Channel Number = 0
                     Target Number = 11
              Device Number = 4
```

```
                Channel Number = 0
                Target Number = 12
        Device Number = 5
                Channel Number = 0
                Target Number = 13
        Device Number = 6
                Channel Number = 0
                Target Number = 14
        Device Number = 7
                Channel Number = 0
                Target Number = 15
SPAN Number = 1
        Starting Block = 0
        Number of blocks = 17913856
        Device Number = 0
                Channel Number = 1
                Target Number = 8
        Device Number = 1
                Channel Number = 1
                Target Number = 9
        Device Number = 2
                Channel Number = 1
                Target Number = 10
        Device Number = 3
                Channel Number = 1
                Target Number = 11
        Device Number = 4
                Channel Number = 1
                Target Number = 12
        Device Number = 5
                Channel Number = 1
                Target Number = 13
        Device Number = 6
                Channel Number = 1
                Target Number = 14
        Device Number = 7
                Channel Number = 1
                Target Number = 15
SPAN Number = 2
        Starting Block = 0
        Number of blocks = 17913856
        Device Number = 0
                Channel Number = 2
                Target Number = 8
        Device Number = 1
                Channel Number = 2
                Target Number = 9
        Device Number = 2
                Channel Number = 2
                Target Number = 10
        Device Number = 3
                Channel Number = 2
                Target Number = 11
        Device Number = 4
                Channel Number = 2
                Target Number = 12
        Device Number = 5
                Channel Number = 2
                Target Number = 13
        Device Number = 6
                Channel Number = 2
                Target Number = 14
        Device Number = 7
                Channel Number = 2
                Target Number = 15
SPAN Number = 3
        Starting Block = 0
        Number of blocks = 17913856
        Device Number = 0
                Channel Number = 3
                Target Number = 8
        Device Number = 1
                Channel Number = 3
                Target Number = 9
        Device Number = 2
                Channel Number = 3
                Target Number = 10
        Device Number = 3
                Channel Number = 3
                Target Number = 11
        Device Number = 4
                Channel Number = 3
                Target Number = 12
        Device Number = 5
                Channel Number = 3
                Target Number = 13
        Device Number = 6
                Channel Number = 3
                Target Number = 14
        Device Number = 7
                Channel Number = 3
                Target Number = 15


(Channel 0, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603
```

```
(Channel 0, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603


Adapter No: 5

Number of Logical Drives : 1

Logical Drive = 0
       Capacity = 279904MB
       Span Depth = 4
       Raid Level = 0,
       Read Ahead = NORMAL
       Stripe Size = 64KB,
       Status = OPTIMAL
       Write Policy = WRITE_THRU,
       Direct IO = DIRECT_IO,
```

```
Number of Stripes = 8
SPAN Number = 0
        Starting Block = 0
        Number of blocks = 17913856
        Device Number = 0
                Channel Number = 0
                Target Number = 8
        Device Number = 1
                Channel Number = 0
                Target Number = 9
        Device Number = 2
                Channel Number = 0
                Target Number = 10
        Device Number = 3
                Channel Number = 0
                Target Number = 11
        Device Number = 4
                Channel Number = 0
                Target Number = 12
        Device Number = 5
                Channel Number = 0
                Target Number = 13
        Device Number = 6
                Channel Number = 0
                Target Number = 14
        Device Number = 7
                Channel Number = 0
                Target Number = 15
SPAN Number = 1
        Starting Block = 0
        Number of blocks = 17913856
        Device Number = 0
                Channel Number = 1
                Target Number = 8
        Device Number = 1
                Channel Number = 1
                Target Number = 9
        Device Number = 2
                Channel Number = 1
                Target Number = 10
        Device Number = 3
                Channel Number = 1
                Target Number = 11
        Device Number = 4
                Channel Number = 1
                Target Number = 12
        Device Number = 5
                Channel Number = 1
                Target Number = 13
        Device Number = 6
                Channel Number = 1
                Target Number = 14
        Device Number = 7
                Channel Number = 1
                Target Number = 15
SPAN Number = 2
        Starting Block = 0
        Number of blocks = 17913856
        Device Number = 0
                Channel Number = 2
                Target Number = 8
        Device Number = 1
                Channel Number = 2
                Target Number = 9
        Device Number = 2
                Channel Number = 2
                Target Number = 10
        Device Number = 3
                Channel Number = 2
                Target Number = 11
        Device Number = 4
                Channel Number = 2
                Target Number = 12
        Device Number = 5
                Channel Number = 2
                Target Number = 13
        Device Number = 6
                Channel Number = 2
                Target Number = 14
        Device Number = 7
                Channel Number = 2
                Target Number = 15
SPAN Number = 3
        Starting Block = 0
        Number of blocks = 17913856
        Device Number = 0
                Channel Number = 3
                Target Number = 8
        Device Number = 1
                Channel Number = 3
                Target Number = 9
        Device Number = 2
                Channel Number = 3
                Target Number = 10
        Device Number = 3
                Channel Number = 3
                Target Number = 11
        Device Number = 4
                Channel Number = 3
                Target Number = 12
        Device Number = 5
                Channel Number = 3
                Target Number = 13
        Device Number = 6
                Channel Number = 3
                Target Number = 14
        Device Number = 7
                Channel Number = 3
                Target Number = 15


(Channel 0, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603
```

```
(Channel 0, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 15)
```

```
        Type = HARDDISK,       Current Status = ONLINE
        Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603


Adapter No: 6

Number of Logical Drives : 1

Logical Drive = 0
        Capacity = 279904MB
        Span Depth = 4
        Raid Level = 0,
        Read Ahead = NORMAL
        Stripe Size = 64KB,
        Status = OPTIMAL
        Write Policy = WRITE_THRU,
        Direct IO = DIRECT_IO,
        Number of Stripes = 8
        SPAN Number = 0
                Starting Block = 0
                Number of blocks = 17913856
                Device Number = 0
                        Channel Number = 0
                        Target Number = 8
                Device Number = 1
                        Channel Number = 0
                        Target Number = 9
                Device Number = 2
                        Channel Number = 0
                        Target Number = 10
                Device Number = 3
                        Channel Number = 0
                        Target Number = 11
                Device Number = 4
                        Channel Number = 0
                        Target Number = 12
                Device Number = 5
                        Channel Number = 0
                        Target Number = 13
                Device Number = 6
                        Channel Number = 0
                        Target Number = 14
                Device Number = 7
                        Channel Number = 0
                        Target Number = 15
        SPAN Number = 1
                Starting Block = 0
                Number of blocks = 17913856
                Device Number = 0
                        Channel Number = 1
                        Target Number = 8
                Device Number = 1
                        Channel Number = 1
                        Target Number = 9
                Device Number = 2
                        Channel Number = 1
                        Target Number = 10
                Device Number = 3
                        Channel Number = 1
                        Target Number = 11
                Device Number = 4
                        Channel Number = 1
                        Target Number = 12
                Device Number = 5
                        Channel Number = 1
                        Target Number = 13
                Device Number = 6
                        Channel Number = 1
                        Target Number = 14
                Device Number = 7
                        Channel Number = 1
                        Target Number = 15
        SPAN Number = 2
                Starting Block = 0
                Number of blocks = 17913856
                Device Number = 0
                        Channel Number = 2
                        Target Number = 8
                Device Number = 1
                        Channel Number = 2
                        Target Number = 9
                Device Number = 2
                        Channel Number = 2
                        Target Number = 10
                Device Number = 3
                        Channel Number = 2
                        Target Number = 11
                Device Number = 4
                        Channel Number = 2
                        Target Number = 12
                Device Number = 5
                        Channel Number = 2
                        Target Number = 13
                Device Number = 6
                        Channel Number = 2
                        Target Number = 14
                Device Number = 7
                        Channel Number = 2
                        Target Number = 15
        SPAN Number = 3
                Starting Block = 0
                Number of blocks = 17913856
                Device Number = 0
                        Channel Number = 3
                        Target Number = 8
                Device Number = 1
                        Channel Number = 3
                        Target Number = 9
                Device Number = 2
                        Channel Number = 3
                        Target Number = 10
                Device Number = 3
                        Channel Number = 3
                        Target Number = 11
                Device Number = 4
                        Channel Number = 3
                        Target Number = 12
                Device Number = 5
```

```
                Channel Number = 3
                Target Number = 13
          Device Number = 6
                Channel Number = 3
                Target Number = 14
          Device Number = 7
                Channel Number = 3
                Target Number = 15


(Channel 0, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603
```

```
 (Channel 3, ID 11)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 12)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 13)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 14)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

 (Channel 3, ID 15)
       Type = HARDDISK,       Current Status = ONLINE
       Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603


Adapter No: 7

Number of Logical Drives : 1

Logical Drive = 0
       Capacity = 279904MB
       Span Depth = 4
       Raid Level = 0,
       Read Ahead = NORMAL
       Stripe Size = 64KB,
       Status = OPTIMAL
       Write Policy = WRITE_THRU,
       Direct IO = DIRECT_IO,
       Number of Stripes = 8
       SPAN Number = 0
              Starting Block = 0
              Number of blocks = 17913856
              Device Number = 0
                     Channel Number = 0
                     Target Number = 8
              Device Number = 1
                     Channel Number = 0
                     Target Number = 9
              Device Number = 2
                     Channel Number = 0
                     Target Number = 10
              Device Number = 3
                     Channel Number = 0
                     Target Number = 11
              Device Number = 4
                     Channel Number = 0
                     Target Number = 12
              Device Number = 5
                     Channel Number = 0
                     Target Number = 13
              Device Number = 6
                     Channel Number = 0
                     Target Number = 14
              Device Number = 7
                     Channel Number = 0
                     Target Number = 15
       SPAN Number = 1
              Starting Block = 0
              Number of blocks = 17913856
              Device Number = 0
                     Channel Number = 1
                     Target Number = 8
              Device Number = 1
                     Channel Number = 1
                     Target Number = 9
              Device Number = 2
                     Channel Number = 1
                     Target Number = 10
              Device Number = 3
                     Channel Number = 1
                     Target Number = 11
              Device Number = 4
                     Channel Number = 1
                     Target Number = 12
              Device Number = 5
                     Channel Number = 1
                     Target Number = 13
              Device Number = 6
                     Channel Number = 1
                     Target Number = 14
              Device Number = 7
                     Channel Number = 1
                     Target Number = 15
       SPAN Number = 2
              Starting Block = 0
              Number of blocks = 17913856
              Device Number = 0
                     Channel Number = 2
                     Target Number = 8
              Device Number = 1
                     Channel Number = 2
                     Target Number = 9
              Device Number = 2
                     Channel Number = 2
                     Target Number = 10
              Device Number = 3
                     Channel Number = 2
                     Target Number = 11
              Device Number = 4
                     Channel Number = 2
                     Target Number = 12
              Device Number = 5
                     Channel Number = 2
                     Target Number = 13
              Device Number = 6
                     Channel Number = 2
                     Target Number = 14
              Device Number = 7
                     Channel Number = 2
                     Target Number = 15
       SPAN Number = 3
              Starting Block = 0
```

```
                Number of blocks = 17913856
                Device Number = 0
                       Channel Number = 3
                       Target Number = 8
                Device Number = 1
                       Channel Number = 3
                       Target Number = 9
                Device Number = 2
                       Channel Number = 3
                       Target Number = 10
                Device Number = 3
                       Channel Number = 3
                       Target Number = 11
                Device Number = 4
                       Channel Number = 3
                       Target Number = 12
                Device Number = 5
                       Channel Number = 3
                       Target Number = 13
                Device Number = 6
                       Channel Number = 3
                       Target Number = 14
                Device Number = 7
                       Channel Number = 3
                       Target Number = 15


(Channel 0, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 0, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 1, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 2, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603
```

```
(Channel 2, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 8)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 9)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 10)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 11)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 12)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 13)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 14)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603

(Channel 3, ID 15)
      Type = HARDDISK,       Current Status = ONLINE
      Size 17913856 blocks,  8747MB  UNISYS  007114ST39102LC  B603
```

External RAID Configuration Parameters

```
************************************************************************
*        Unisys Ultra-Wide RAID Controller   OSM1200-RAD               *
************************************************************************

CPU type:      5X86-133(WB)
Serial Number  3056383
Firmware version 2.22V
Bootcode version 1.12G
Total cache:   32MB (EDO)

- Cache        Write Back:           enabled
               optimization:         sequential (128K stripe size)

- Raid         Rebuild Priority:     low
               Write Priority        on Initialization:    disabled
                                     on Rebuild:           disabled
                                     on Normal:            disabled

- SCSI         SCSI Motor Spin-Up    Disable
               Power-Up SCSI Reset   Enable
               SMART                 Disabled


Physical Drives

       Slot   Chl   Id    Capacity   Status   XferRate    Vendor/Product
Id
               1     8     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603
               1     9     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603
               1    10     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603
               1    11     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603
               1    12     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603
               1    13     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603
               1    14     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603
               1    15     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603


Logical Drives

         ID     RAID Level  Capacity  Status  # Drives  # On-line  # Spares
# Failed
       D02756E    RAID 0     69984MB  Good       8          8        0
          0


Volume

               Logical Volume Partition table
               ID D02756E     Capacity 69984 MB

       Idx     Base Offset    Part Size
        0         0 MB        69984 MB


Host LUN Assignment

       SCSI Chl       LUN     LVIDx   PortIdx   Capacity
           2           0       0         0     69984 MB



************************************************************************
*        Unisys Ultra-Wide RAID Controller   OSM1200-RAD               *
************************************************************************

CPU type:      5X86-133(WB)
```

```
Serial Number  3056390
Firmware version 2.22V
Bootcode version 1.12G
Total cache:   32MB (EDO)

- Cache        Write Back:           enabled
               optimization:         sequential (128K stripe size)

- Raid         Rebuild Priority:     low
               Write Priority        on Initialization:    disabled
                                     on Rebuild:           disabled
                                     on Normal:            disabled

- SCSI         SCSI Motor Spin-Up    Disable
               Power-Up SCSI Reset   Enable
               SMART                 Disabled


Physical Drives

       Slot   Chl   Id    Capacity   Status   XferRate    Vendor/Product
Id
               1     8     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603
               1     9     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603
               1    10     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603
               1    11     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603
               1    12     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603
               1    13     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603
               1    14     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603
               1    15     8748 MB    online     83 MB     UNISYS
007114ST391022LC B603


Logical Drives

         ID     RAID Level  Capacity  Status  # Drives  # On-line  # Spares
# Failed
       46CE84DF   RAID 0     69984MB Good       8         8        0
          0


Volume

               Logical Volume Partition table
               ID 46CE84DF    Capacity 69984 MB

       Idx     Base Offset    Part Size
        0         0 MB        69984 MB


Host LUN Assignment

       SCSI Chl       LUN     LVIDx   PortIdx   Capacity
           0           0       0         0     69984 MB
```

Configuration of Log Drives

A single MegaRAID Enterprise 1500-H (AMI) controller was used in the SUT for the mirrored log drives. Half of the drives were in one disk cage connected to one channel of the controller and half were in a second disk cage connected to a second channel of the controller. The controller implemented the RAID 1 mirorring across the two channels. Write caching was disabled on both the controller and on all the physical drives themselves.

Two OSM311000-LR disk cages, which include one OSM1200-RAD SCSI-to-SCSI RAID controller each, was used for the log. Each of these controllers implemented RAID 0 striping on the eight 9GB drives that were in each disk cage, so that the Enterprise 1500-H controller in the SUT saw just two large 'disks'. Each of the OSM1200-RAD controllers had a 32MB cache. Configuration options were set for Write Back caching and Optimized for Sequential IO. The OSM1200-RAD controllers used an algorithm that ensured that cached write data was held for no more than a fraction of a minute before being written to the physical drives.

For the priced configuration, each of the disk cages contained two redundant power supplies. Only one was required to be functional to keep the OSM1200-RAD controller and disk drives operational. A UPS was priced to provide power to one power supply in each disk cage. The second power supply in each disk cage was connected to normal wall power. Thus neither interruption of power or failure of the UPS would affect the two log disk cages (or their controllers and disks). Since the two disk cages were completely independent of each other, this configuration ensured that there was no single point of failure in writing to the log.

NT Server Configuration Information

```
Microsoft Diagnostics Report For \\AVALON4
------------------------------------------------------------------------

OS Version Report
------------------------------------------------------------------------
Microsoft (R) Windows NT (TM) Server
Version 4.0 (Build 1381: Service Pack 4) x86 Multiprocessor Free
Registered Owner: SAM&M, Unisys Corporation
Product Number: 70234-810-6895975-67328
```

```
------------------------------------------------------------------------

System Report
------------------------------------------------------------------------
System: AT/AT COMPATIBLE
Hardware Abstraction Layer: MPS 1.4 - APIC platform
BIOS Date: 08/02/99
BIOS Version: OCPRF100- PhoenixBIOS 4.0 Releas

Processor list:
   0:  x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
   1:  x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
   2:  x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
   3:  x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
------------------------------------------------------------------------

Video Display Report
------------------------------------------------------------------------
BIOS Date: 05/29/98
BIOS Version: CL-GD5446 PCI VGA BIOS Version 1.35

Adapter:
   Setting: 1024 x 768 x 256
            75 Hz
   Type: cirrus compatible display adapter
   String: Cirrus Logic Compatible
   Memory: 2 MB
   Chip Type: Cirrus Logic 5446
   DAC Type: Integrated RAMDAC
Driver:
   Vendor: Microsoft Corporation
   File(s): cirrus.sys, vga.dll, cirrus.dll, vga256.dll, vga64K.dll
   Version: 4.00, 4.0.0


Drives Report
------------------------------------------------------------------------
C:\  (Local - FAT) SYSTEM Total: 2,096,160 KB, Free: 896,416 KB
   Serial Number: F035 - 8AA4
   Bytes per cluster: 512
   Sectors per cluster: 64
   Filename length: 255
Z:\  (Local - NTFS) testfiles Total: 2,345,488 KB, Free: 918,272 KB
   Serial Number: B0C5 - 33C8
   Bytes per cluster: 512
   Sectors per cluster: 8
   Filename length: 255

Memory Report
------------------------------------------------------------------------
Handles: 1,146
Threads: 109
Processes: 13

Physical Memory (K)
   Total: 3,931,524
   Available: 3,712,264
   File Cache: 13,308

Kernel Memory (K)
   Total: 11,576
   Paged: 8,724
   Nonpaged: 2,852

Commit Charge (K)
   Total: 75,324
   Limit: 4,433,004
   Peak: 75,348

Pagefile Space (K)
   Total: 655,360
   Total in use: 0
   Peak: 0

   C:\pagefile.sys
      Total: 524,288
      Total in use: 0
      Peak: 0

   Z:\pagefile.sys
      Total: 131,072
      Total in use: 0
      Peak: 0



Services Report
------------------------------------------------------------------------
Adaptec Array Controller Remote Services Agent Stopped   (Disabled)
   C:\Program Files\Adaptec\AAC\System\AfaAgent.Exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process, Interactive
   Service Dependencies:
      RpcSs
Alerter                                          Stopped   (Manual)
   C:\WINNT\System32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Service Dependencies:
      LanmanWorkstation
Computer Browser                                 Stopped   (Manual)
   C:\WINNT\System32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Service Dependencies:
      LanmanWorkstation
      LanmanServer
      LmHosts
ClipBook Server                                  Stopped   (Manual)
   C:\WINNT\system32\clipsrv.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
   Service Dependencies:
      NetDDE
```

```
PCI Hot Plug Service                    Stopped   (Disabled)
   C:\WINNT\System32\cpqphps.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
DHCP Client (TDI)                       Stopped   (Disabled)
   C:\WINNT\System32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Service Dependencies:
      Tcpip
      Afd
      NetBT
EventLog (Event log)                    Running   (Automatic)
   C:\WINNT\system32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
Server                                  Running   (Automatic)
   C:\WINNT\System32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Group Dependencies:
      TDI
Workstation (NetworkProvider)           Running   (Automatic)
   C:\WINNT\System32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Group Dependencies:
      TDI
License Logging Service                 Stopped   (Manual)
   C:\WINNT\System32\llssrv.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
TCP/IP NetBIOS Helper                   Stopped   (Manual)
   C:\WINNT\System32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Group Dependencies:
      NetworkProvider
MegaServ                                Stopped   (Automatic)
   C:\WINNT\system32\megaserv.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
Messenger                               Stopped   (Manual)
   C:\WINNT\System32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Service Dependencies:
      LanmanWorkstation
      NetBios
MSDTC (MS Transactions)                 Stopped   (Disabled)
   C:\WINNT\System32\msdtc.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
   Service Dependencies:
      RPCSS
      NTLMSSP
MSSQLServer                             Stopped   (Manual)
   C:\MSSQL7\binn\sqlservr.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
Network DDE (NetDDEGroup)               Stopped   (Disabled)
   C:\WINNT\system32\netdde.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Service Dependencies:
      NetDDEDSDM
Network DDE DSDM                        Stopped   (Disabled)
   C:\WINNT\system32\netdde.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
Net Logon (RemoteValidation)            Stopped   (Manual)
   C:\WINNT\System32\lsass.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Service Dependencies:
      LanmanWorkstation
      LmHosts
NT LM Security Support Provider         Stopped   (Manual)
   C:\WINNT\System32\SERVICES.EXE
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
Plug and Play (PlugPlay)                Running   (Automatic)
   C:\WINNT\system32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
Protected Storage                       Running   (Automatic)
   C:\WINNT\System32\pstores.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process, Interactive
   Service Dependencies:
      RpcSs
Directory Replicator                    Stopped   (Manual)
   C:\WINNT\System32\lmrepl.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
   Service Dependencies:
      LanmanWorkstation
      LanmanServer
Remote Procedure Call (RPC) Locator     Stopped   (Manual)
   C:\WINNT\System32\LOCATOR.EXE
```

```
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
   Service Dependencies:
      LanmanWorkstation
      Rdr
Remote Procedure Call (RPC) Service          Running   (Manual)
   C:\WINNT\system32\RpcSs.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
Schedule                                     Stopped   (Manual)
   C:\WINNT\System32\AtSvc.Exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
SNMP                                         Stopped   (Disabled)
   C:\WINNT\System32\snmp.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
   Service Dependencies:
      Tcpip
      EventLog
SNMP Trap Service                            Stopped   (Disabled)
   C:\WINNT\System32\snmptrap.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
   Service Dependencies:
      Tcpip
      EventLog
Spooler (SpoolerGroup)                       Stopped   (Manual)
   C:\WINNT\system32\spoolss.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process, Interactive
SQLServerAgent                               Stopped   (Manual)
   C:\MSSQL7\binn\sqlagent.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
   Service Dependencies:
      MSSQLServer
Telephony Service                            Stopped   (Manual)
   C:\WINNT\system32\tapisrv.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
UPS                                          Stopped   (Manual)
   C:\WINNT\System32\ups.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process



Drivers Report
--------------------------------------------------------------------------
```

```
aaccomm (SCSI miniport)                      Stopped   (Disabled)
   C:\WINNT\System32\DRIVERS\aaccomm.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
aacdisk (Primary Disk)                       Stopped   (Disabled)
   C:\WINNT\System32\DRIVERS\aacdisk.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
aacport (SCSI miniport)                      Stopped   (Disabled)
   C:\WINNT\System32\DRIVERS\aacport.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
aacscsi (port)                               Stopped   (Disabled)
   C:\WINNT\System32\DRIVERS\aacscsi.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
      SCSI miniport
Abiosdsk (Primary disk)                      Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
AFD Networking Support Environment (TDI)     Running   (Automatic)
   C:\WINNT\System32\drivers\afd.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Aha154x (SCSI miniport)                      Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Aha174x (SCSI miniport)                      Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
aic78xx (SCSI miniport)                      Stopped   (Boot)
   C:\WINNT\System32\DRIVERS\aic78xx.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Always (SCSI miniport)                       Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
ami0nt (SCSI miniport)                       Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
amsint (SCSI miniport)                       Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Arrow (SCSI miniport)                        Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
atapi (SCSI miniport)                        Running   (Boot)
   C:\WINNT\System32\DRIVERS\atapi.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Atdisk (Primary disk)                        Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
ati (Video)                                  Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Beep (Base)                                  Running   (System)
   Error Severity: Normal
```

```
   Service Flags: Kernel Driver, Shared Process
BusLogic (SCSI miniport)                    Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Busmouse (Pointer Port)                     Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Cdaudio (Filter)                            Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Cdfs (File system)                          Running   (Disabled)
   Error Severity: Normal
   Service Flags: File System Driver, Shared Process
   Group Dependencies:
      SCSI CDROM Class
Cdrom (SCSI CDROM Class)                    Running   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
      SCSI miniport
Changer (Filter)                            Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
cirrus (Video)                              Running   (System)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Cpqarray (SCSI miniport)                    Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
cpqfws2e (SCSI miniport)                    Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
dac960nt (SCSI miniport)                    Stopped   (Disabled)
   C:\WINNT\System32\drivers\dac960nt.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
dce376nt (SCSI miniport)                    Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Delldsa (SCSI miniport)                     Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Dell_DGX (Video)                            Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Disk (SCSI Class)                           Running   (Boot)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
      SCSI miniport
Diskperf (Filter)                           Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
DptScsi (SCSI miniport)                     Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
dtc329x (SCSI miniport)                     Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Intel(R) PRO NDIS Driver (NDIS)             Running   (Automatic)
   C:\WINNT\System32\drivers\E100BNT.SYS
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
em (Base)                                   Running   (System)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
et4000 (Video)                              Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Fastfat (Boot file system)                  Running   (Disabled)
   Error Severity: Normal
   Service Flags: File System Driver, Shared Process
Fd16_700 (SCSI miniport)                    Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Fd7000ex (SCSI miniport)                    Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Fd8xx (SCSI miniport)                       Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
flashpnt (SCSI miniport)                    Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Floppy (Primary disk)                       Running   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Ftdisk (Filter)                             Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
gamdrv (SCSI Class)                         Stopped   (Manual)
   C:\WINNT\System32\drivers\gamdrv.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
PCI Hot Plug Driver                         Stopped   (Disabled)
   System32\DRIVERS\hotplug.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
i8042 Keyboard and PS/2 Mouse Port Driver (Keyboard Port) Running
(System)
   System32\DRIVERS\i8042prt.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Inport (Pointer Port)                       Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Jazzg300 (Video)                            Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Jazzg364 (Video)                            Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Jzvxl484 (Video)                            Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Keyboard Class Driver (Keyboard Class)      Running   (System)
   System32\DRIVERS\kbdclass.sys
   Error Severity: Normal
```

```
   Service Flags: Kernel Driver, Shared Process
KSecDD (Base)                                 Running   (System)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
macdisk (Filter)                              Stopped   (Disabled)
   C:\WINNT\System32\drivers\macdisk.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
MEGARAID (SCSI miniport)                      Stopped   (Disabled)
   C:\WINNT\System32\drivers\megaraid.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
mga (Video)                                   Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
mga_mil (Video)                               Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
mitsumi (SCSI miniport)                       Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
mkecr5xx (SCSI miniport)                      Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Modem (Extended base)                         Stopped   (Manual)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Mouse Class Driver (Pointer Class)            Running   (System)
   System32\DRIVERS\mouclass.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
mraid (Primary disk)                          Running   (Boot)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Msfs (File system)                            Running   (System)
   Error Severity: Normal
   Service Flags: File System Driver, Shared Process
Mup (Network)                                 Running   (Manual)
   C:\WINNT\System32\drivers\mup.sys
   Error Severity: Normal
   Service Flags: File System Driver, Shared Process
NetBEUI Protocol (PNP_TDI)                    Running   (Automatic)
   C:\WINNT\System32\drivers\nbf.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Ncr53c9x (SCSI miniport)                      Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
ncr77c22 (Video)                              Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Ncrc700 (SCSI miniport)                       Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Ncrc710 (SCSI miniport)                       Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Microsoft NDIS System Driver (NDIS)           Running   (System)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
NetBIOS Interface (NetBIOSGroup)              Stopped   (Manual)
   C:\WINNT\System32\drivers\netbios.sys
   Error Severity: Normal
   Service Flags: File System Driver, Shared Process
   Group Dependencies:
      TDI
WINS Client(TCP/IP) (PNP_TDI)                 Stopped   (Disabled)
   C:\WINNT\System32\drivers\netbt.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
   Service Dependencies:
      Tcpip
NetDetect                                     Stopped   (Manual)
   C:\WINNT\system32\drivers\netdtect.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Npfs (File system)                            Running   (System)
   Error Severity: Normal
   Service Flags: File System Driver, Shared Process
Ntfs (File system)                            Running   (Disabled)
   Error Severity: Normal
   Service Flags: File System Driver, Shared Process
Null (Base)                                   Running   (System)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Oliscsi (SCSI miniport)                       Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Parallel (Extended base)                      Stopped   (Manual)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Service Dependencies:
      Parport
   Group Dependencies:
      Parallel arbitrator
Parport (Parallel arbitrator)                 Stopped   (Manual)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
ParVdm (Extended base)                        Stopped   (Manual)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Service Dependencies:
      Parport
   Group Dependencies:
      Parallel arbitrator
PCIDump (PCI Configuration)                   Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Pcmcia (System Bus Extender)                  Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
PnP ISA Enabler Driver (Base)                 Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
PortFltr (port)                               Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
```

```
      SCSI miniport
psidisp (Video)                                 Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Ql10wnt (SCSI miniport)                         Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
qv (Video)                                      Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Rdr (Network)                                   Running   (Manual)
   C:\WINNT\System32\drivers\rdr.sys
   Error Severity: Normal
   Service Flags: File System Driver, Shared Process
s3 (Video)                                      Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Scsiprnt (Extended base)                        Stopped   (Automatic)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
      SCSI miniport
Scsiscan (SCSI Class)                           Running   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
      SCSI miniport
Serial (Extended base)                          Running   (Automatic)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Sermouse (Pointer Port)                         Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Sfloppy (Primary disk)                          Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
      SCSI miniport
Simbad (Filter)                                 Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
slcd32 (SCSI miniport)                          Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Sparrow (SCSI miniport)                         Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Spock (SCSI miniport)                           Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Srv (Network)                                   Running   (Manual)
   C:\WINNT\System32\drivers\srv.sys
   Error Severity: Normal
   Service Flags: File System Driver, Shared Process
symc810 (SCSI miniport)                         Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Symc8XX (SCSI Miniport)                         Stopped   (Boot)
   C:\WINNT\System32\drivers\Symc8XX.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Sym_hi (SCSI Miniport)                          Running   (Boot)
   C:\WINNT\System32\drivers\Sym_hi.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Sysdrv (Extended Base)                          Stopped   (Manual)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
T128 (SCSI miniport)                            Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
T13B (SCSI miniport)                            Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
TCP/IP Service (PNP_TDI)                        Running   (Automatic)
   C:\WINNT\System32\drivers\tcpip.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
tga (Video)                                     Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
tmv1 (SCSI miniport)                            Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Ultra124 (SCSI miniport)                        Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Ultra14f (SCSI miniport)                        Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Ultra24f (SCSI miniport)                        Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
update (Base)                                   Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
v7vram (Video)                                  Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
VgaSave (Video Save)                            Stopped   (System)
   C:\WINNT\System32\drivers\vga.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
VgaStart (Video Init)                           Stopped   (System)
   C:\WINNT\System32\drivers\vga.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Wd33c93 (SCSI miniport)                         Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
wd90c24a (Video)                                Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
wdvga (Video)                                   Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
weitekp9 (Video)                                Stopped   (Disabled)
   Error Severity: Ignore
```

```
   Service Flags: Kernel Driver, Shared Process
Xga (Video)                                  Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process



IRQ and Port Report
-------------------------------------------------------------------------
Devices                          Vector Level  Affinity
-------------------------------------------------------------------------
MPS 1.4 - APIC platform               8     8 0x0000000f
MPS 1.4 - APIC platform               0     0 0x0000000f
MPS 1.4 - APIC platform               1     1 0x0000000f
MPS 1.4 - APIC platform               2     2 0x0000000f
MPS 1.4 - APIC platform               3     3 0x0000000f
MPS 1.4 - APIC platform               4     4 0x0000000f
MPS 1.4 - APIC platform               5     5 0x0000000f
MPS 1.4 - APIC platform               6     6 0x0000000f
MPS 1.4 - APIC platform               7     7 0x0000000f
MPS 1.4 - APIC platform               8     8 0x0000000f
MPS 1.4 - APIC platform               9     9 0x0000000f
MPS 1.4 - APIC platform              10    10 0x0000000f
MPS 1.4 - APIC platform              11    11 0x0000000f
MPS 1.4 - APIC platform              12    12 0x0000000f
MPS 1.4 - APIC platform              13    13 0x0000000f
MPS 1.4 - APIC platform              14    14 0x0000000f
MPS 1.4 - APIC platform              15    15 0x0000000f
MPS 1.4 - APIC platform              16    16 0x0000000f
MPS 1.4 - APIC platform              17    17 0x0000000f
MPS 1.4 - APIC platform              18    18 0x0000000f
MPS 1.4 - APIC platform              19    19 0x0000000f
MPS 1.4 - APIC platform              20    20 0x0000000f
MPS 1.4 - APIC platform              21    21 0x0000000f
MPS 1.4 - APIC platform              22    22 0x0000000f
MPS 1.4 - APIC platform              23    23 0x0000000f
MPS 1.4 - APIC platform              24    24 0x0000000f
MPS 1.4 - APIC platform              25    25 0x0000000f
MPS 1.4 - APIC platform              26    26 0x0000000f
MPS 1.4 - APIC platform              27    27 0x0000000f
MPS 1.4 - APIC platform              28    28 0x0000000f
MPS 1.4 - APIC platform              29    29 0x0000000f
MPS 1.4 - APIC platform              30    30 0x0000000f
MPS 1.4 - APIC platform              31    31 0x0000000f
MPS 1.4 - APIC platform              32    32 0x0000000f
MPS 1.4 - APIC platform              33    33 0x0000000f
MPS 1.4 - APIC platform              34    34 0x0000000f
MPS 1.4 - APIC platform              35    35 0x0000000f
MPS 1.4 - APIC platform              36    36 0x0000000f
MPS 1.4 - APIC platform              37    37 0x0000000f
MPS 1.4 - APIC platform              38    38 0x0000000f
MPS 1.4 - APIC platform              39    39 0x0000000f
MPS 1.4 - APIC platform              40    40 0x0000000f
MPS 1.4 - APIC platform              41    41 0x0000000f
MPS 1.4 - APIC platform              42    42 0x0000000f
MPS 1.4 - APIC platform              43    43 0x0000000f
MPS 1.4 - APIC platform              44    44 0x0000000f
MPS 1.4 - APIC platform              45    45 0x0000000f
MPS 1.4 - APIC platform              46    46 0x0000000f
MPS 1.4 - APIC platform              47    47 0x0000000f
MPS 1.4 - APIC platform              61    61 0x0000000f
MPS 1.4 - APIC platform              65    65 0x0000000f
MPS 1.4 - APIC platform              80    80 0x0000000f
MPS 1.4 - APIC platform             193   193 0x0000000f
MPS 1.4 - APIC platform             225   225 0x0000000f
MPS 1.4 - APIC platform             253   253 0x0000000f
MPS 1.4 - APIC platform             254   254 0x0000000f
MPS 1.4 - APIC platform             255   255 0x0000000f
i8042prt                              1     1 0xffffffff
i8042prt                             12    12 0xffffffff
Serial                                4     4 0x00000000
Serial                                3     3 0x00000000
E100B                                16    16 0x6048102b
Floppy                                6     6 0x00000000
atapi                                 0    14 0x00000000
dac960nt                             32    32 0x00000000
Sym_hi                               40    40 0x00000000
Sym_hi                               41    41 0x00000000
-------------------------------------------------------------------------
Devices                        Physical Address  Length
-------------------------------------------------------------------------
MPS 1.4 - APIC platform         0x00000000  0x0000000010
MPS 1.4 - APIC platform         0x00000020  0x0000000002
MPS 1.4 - APIC platform         0x00000040  0x0000000004
MPS 1.4 - APIC platform         0x00000048  0x0000000004
MPS 1.4 - APIC platform         0x00000061  0x0000000001
MPS 1.4 - APIC platform         0x00000070  0x0000000002
MPS 1.4 - APIC platform         0x00000080  0x0000000010
MPS 1.4 - APIC platform         0x00000092  0x0000000001
MPS 1.4 - APIC platform         0x000000a0  0x0000000002
MPS 1.4 - APIC platform         0x000000c0  0x0000000010
MPS 1.4 - APIC platform         0x000000f0  0x0000000010
i8042prt                        0x00000060  0x0000000001
i8042prt                        0x00000064  0x0000000001
Serial                          0x000003f8  0x0000000007
Serial                          0x000002f8  0x0000000007
E100B                           0x00001800  0x000000001e
Floppy                          0x000003f0  0x0000000006
Floppy                          0x000003f7  0x0000000001
atapi                           0x000001f0  0x0000000008
atapi                           0x000003f6  0x0000000001
dac960nt                        0x00002000  0x0000000080
Sym_hi                          0x00001000  0x0000000100
Sym_hi                          0x00001400  0x0000000100
cirrus                          0x000003b0  0x000000000c
cirrus                          0x000003c0  0x0000000020



DMA and Memory Report
-------------------------------------------------------------------------
Devices                        Channel    Port
-------------------------------------------------------------------------
Floppy                           2     0
-------------------------------------------------------------------------
Devices                        Physical Address  Length
-------------------------------------------------------------------------
MPS 1.4 - APIC platform         0xfec00000  0x00000400
```

```
MPS 1.4 - APIC platform         0xfee00000  0x00000400
E100B                           0xf4400000  0x0000001e
dac960nt                        0xf8110000  0x00000080
dac960nt                        0xfa000000  0x02000000
Sym_hi                          0xf4005400  0x00000400
Sym_hi                          0xf4000000  0x00002000
Sym_hi                          0xf4005800  0x00000400
Sym_hi                          0xf4002000  0x00002000
cirrus                          0x000a0000  0x00020000
cirrus                          0xf6000000  0x02000000



Environment Report
------------------------------------------------------------------------



System Environment Variables
   ComSpec=C:\WINNT\system32\cmd.exe
   HOME=C:/
   NTRESKIT=Z:\NTRESKIT
   NUMBER_OF_PROCESSORS=4
   OS=Windows_NT
   Os2LibPath=C:\WINNT\system32\os2\dll;

Path=C:\MKS\mksnt;C:\WINNT\system32;C:\WINNT;Z:\NTRESKIT;Z:\NTRESKIT\Perl;
C:\MSSQL7\BINN;z:\intel\emon\bin;C:\Program Files\Adaptec\AAC\System
   PROCESSOR_ARCHITECTURE=x86
   PROCESSOR_IDENTIFIER=x86 Family 6 Model 7 Stepping 3, GenuineIntel
   PROCESSOR_LEVEL=6
   PROCESSOR_REVISION=0703
   ROOTDIR=C:/MKS
   SHELL=C:/MKS/mksnt/sh.exe
   TMPDIR=C:/TEMP
   windir=C:\WINNT



Environment Variables for Current User
   TEMP=C:\TEMP
   TMP=C:\TEMP



Network Report
------------------------------------------------------------------------
Your Access Level: Admin & Local
Workgroup or Domain: WORKGROUP
Network Version: 4.0
LanRoot: WORKGROUP
Logged On Users: 1
Current User (1): Administrator
  Logon Domain: AVALON4
  Logon Server: AVALON4

Transport: Nbf_E100B1, 00-A0-C9-DF-2C-E5, VC's: 0, Wan: Wan

Character Wait: 3,600
Collection Time: 250
Maximum Collection Count: 16
Keep Connection: 600
Maximum Commands: 5
Session Time Out: 45
Character Buffer Size: 512
Maximum Threads: 17
Lock Quota: 6,144
Lock Increment: 10
Maximum Locks: 500
Pipe Increment: 10
Maximum Pipes: 500
Cache Time Out: 40
Dormant File Limit: 45
Read Ahead Throughput: 4,294,967,295
Mailslot Buffers: 3
Server Announce Buffers: 20
Illegal Datagrams: 5
Datagram Reset Frequency: 60
Log Election Packets: False
Use Opportunistic Locking: True
Use Unlock Behind: True
Use Close Behind: True
Buffer Pipes: True
Use Lock, Read, Unlock: True
Use NT Caching: True
Use Raw Read: True
Use Raw Write: True
Use Write Raw Data: True
Use Encryption: True
Buffer Deny Write Files: True
Buffer Read Only Files: True
Force Core Creation: True
512 Byte Max Transfer: False
Bytes Received: 269
SMB's Received: 3
Paged Read Bytes Requested: 0
Non Paged Read Bytes Requested: 0
Cache Read Bytes Requested: 0
Network Read Bytes Requested: 0
Bytes Transmitted: 485
SMB's Transmitted: 3
Paged Read Bytes Requested: 0
Non Paged Read Bytes Requested: 0
Cache Read Bytes Requested: 0
Network Read Bytes Requested: 0
Initally Failed Operations: 0
Failed Completion Operations: 0
Read Operations: 0
Random Read Operations: 0
Read SMB's: 0
Large Read SMB's: 0
Small Read SMB's: 0
Write Operations: 0
Random Write Operations: 0
Write SMB's: 0
Large Write SMB's: 0
Small Write SMB's: 0
Raw Reads Denied: 0
```

```
Raw Writes Denied: 0
Network Errors: 0
Sessions: 1
Failed Sessions: 0
Reconnects: 0
Core Connects: 0
LM 2.0 Connects: 0
LM 2.x Connects: 0
Windows NT Connects: 1
Server Disconnects: 0
Hung Sessions: 0
Use Count: 0
Failed Use Count: 0
Current Commands: 0
Server File Opens: 0
Server Device Opens: 0
Server Jobs Queued: 0
Server Session Opens: 0
Server Sessions Timed Out: 0
Server Sessions Errored Out: 0
Server Password Errors: 0
Server Permission Errors: 0
Server System Errors: 0
Server Bytes Sent: 269
Server Bytes Received: 485
Server Average Response Time: 0
Server Request Buffers Needed: 0
Server Big Buffers Needed: 0
```

NT Server Registry Information

Software\Microsoft\MSDTC

```
Key Name:          SOFTWARE\Microsoft\MSDTC
Class Name:        <NO CLASS>
Last Write Time:   6/16/99 - 4:23 PM
Value 0
  Name:            MaxLogSize
  Type:            REG_DWORD
  Data:            0x200


Key Name:          SOFTWARE\Microsoft\MSDTC\Setup
Class Name:        <NO CLASS>
Last Write Time:   6/16/99 - 4:23 PM
Value 0
  Name:            InstallCode
  Type:            REG_DWORD
  Data:            0

Value 1
  Name:            InstallState
  Type:            REG_DWORD
  Data:            0x1

Value 2
  Name:            MajorVersion
  Type:            REG_DWORD
  Data:            0x20000

Value 3
  Name:            MinorVersion
  Type:            REG_DWORD
  Data:            0x32a


Key Name:          SOFTWARE\Microsoft\MSDTC\Setup\ExitStatus
Class Name:        <NO CLASS>
Last Write Time:   6/16/99 - 4:23 PM
Value 0
  Name:            CompletionComment
  Type:            REG_SZ
  Data:            Source = DtcComplete, ExitType = Success, Successful
Install

Value 1
  Name:            ErrorCode
  Type:            REG_DWORD
  Data:            0

Value 2
  Name:            ExitCode
  Type:            REG_DWORD
  Data:            0

Value 3
  Name:            Source
  Type:            REG_DWORD
  Data:            0x1
```

Software\Microsoft\MSSQLServer

```
Key Name:          SOFTWARE\Microsoft\MSSQLServer
Class Name:        <NO CLASS>
Last Write Time:   11/25/98 - 5:12 PM
Value 0
  Name:            ConfigurationInformation
  Type:            REG_BINARY
  Data:
00000000   ff 9c 5c 36 07 00 00 00 - 41 00 56 00 41 00 4c 00
..\6....A.V.A.L.
00000010   4f 00 4e 00 34 00 00 00 - 00 00 00 00 00 00 00 00
O.N.4...........
00000020   00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
................
00000030   00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
................
00000040   00 00 00 00 00 00 00 00 - 53 00 41 00 4d 00 26 00
........S.A.M.&.
```

```
00000050   4d 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
M...............
00000060   00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
................
00000070   00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
................
00000080   00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
................
00000090   00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
................
000000a0   00 00 00 00 00 00 00 00 - 00 00 00 00 07 00 00 00
................
000000b0   00 00 00 00 6a 02 00 00 - 03 00 00 00 01 00 00 00
....j...........
000000c0   32 00 00 00 00 82 00 00 - 04 00 00 00 04 00 00 00
2...............
000000d0   00 00 00 00 65 05 00 00 - 04 00 00 00 04 00 00 00
....e...........
000000e0   5f 0e 00 00 aa 0d 00 00 - 11 00 00 00             _...........


Key Name:          SOFTWARE\Microsoft\MSSQLServer\Client
Class Name:        <NO CLASS>
Last Write Time:   9/2/98 - 3:13 PM

Key Name:          SOFTWARE\Microsoft\MSSQLServer\Client\ConnectTo
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            DSQUERY
  Type:            REG_SZ
  Data:            DBMSSOCN


Key Name:          SOFTWARE\Microsoft\MSSQLServer\Client\DB-Lib
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            AutoAnsiToOem
  Type:            REG_SZ
  Data:            ON

Value 1
  Name:            UseIntlSettings
  Type:            REG_SZ
  Data:            ON


Key Name:          SOFTWARE\Microsoft\MSSQLServer\Client\TDS
Class Name:        <NO CLASS>
Last Write Time:   5/25/99 - 4:08 PM
Value 0
  Name:            <NO NAME>
  Type:            REG_SZ
  Data:            7.0

Value 1
  Name:            (local)
  Type:            REG_SZ
  Data:            7.0

Value 2
  Name:            .
  Type:            REG_SZ
  Data:            7.0

Value 3
  Name:            Avalon4
  Type:            REG_SZ
  Data:            7.0

Value 4
  Name:            TheLocalServerUsingPipes
  Type:            REG_SZ
  Data:            7.0


Key Name:          SOFTWARE\Microsoft\MSSQLServer\MSSQLServer
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            AuditLevel
  Type:            REG_DWORD
  Data:            0

Value 1
  Name:            BackupDirectory
  Type:            REG_SZ
  Data:            C:\MSSQL7\BACKUP

Value 2
  Name:            DefaultCompStyle
  Type:            REG_SZ
  Data:            0

Value 3
  Name:            DefaultDomain
  Type:            REG_SZ
  Data:            AVALON4

Value 4
  Name:            DefaultLocaleID
  Type:            REG_SZ
  Data:            8200

Value 5
  Name:            DefaultLogin
  Type:            REG_SZ
  Data:            guest

Value 6
  Name:            DefaultSortID
  Type:            REG_SZ
  Data:            50

Value 7
  Name:            ListenOn
  Type:            REG_MULTI_SZ
```

```
  Data:            SSNMPN70,\\.\pipe\sql\query
                   SSMSSO70,1433


Value 8
  Name:            LoginMode
  Type:            REG_DWORD
  Data:            0

Value 9
  Name:            Map#
  Type:            REG_SZ
  Data:            -

Value 10
  Name:            Map$
  Type:            REG_SZ
  Data:

Value 11
  Name:            Map_
  Type:            REG_SZ
  Data:            \

Value 12
  Name:            ResourceMgrID
  Type:            REG_SZ
  Data:            {E5ADE8B6-A98B-11D2-BA85-00A0C9C545C4}

Value 13
  Name:            RWSListenAddress
  Type:            REG_SZ
  Data:

Value 14
  Name:            SetHostName
  Type:            REG_DWORD
  Data:            0

Value 15
  Name:            Tapeloadwaittime
  Type:            REG_DWORD
  Data:            0xffffffff


Key Name:
SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\CurrentVersion
Class Name:        <NO CLASS>
Last Write Time:   6/16/99 - 4:23 PM
Value 0
  Name:            checksum
  Type:            REG_BINARY
  Data:
00000000   37 36 32 32 63 31 35 38 - 61 65 37 64 34 63 64 37
7622c158ae7d4cd7
00000010   35 30 64 61 30 33 34 62 - 37 64 63 33 37 64 64 38
50da034b7dc37dd8
00000020   63 61 65 35 32 62 31 62 - 64 34 38 39 66 33 35 63
cae52b1bd489f35c
00000030   65 64 64 34 32 65 65 61 - 31 65 39 30 32 35 65 63
edd42eea1e9025ec
00000040   39 65 63 31 61 35 38 38 - 62 32 62 65 65 32 61 66
9ec1a588b2bee2af
00000050   37 37 33 63 63 36 38 39 - 37 65 62 39 35 36 31 33
773cc6897eb95613
00000060   65 34 65 31 37 36 33 64 - 33 61 30 66 65 34 63 31
e4e1763d3a0fe4c1
00000070   61 66 38 36 62 64 30 38 - 32 66 62 66 39 38 36 37
af86bd082fbf9867
00000080   38 30 65 61 30 38 31 64 - 32 37 30 63 66 36 64 32
80ea081d270cf6d2
00000090   30 61 34 38 39 63 33 64 - 34 65 00                0a489c3d4e.

Value 1
  Name:            CSDVersion
  Type:            REG_SZ
  Data:            7.00.805

Value 2
  Name:            CSDVersionNumber
  Type:            REG_DWORD
  Data:            0x100

Value 3
  Name:            CurrentVersion
  Type:            REG_SZ
  Data:            7.00.804

Value 4
  Name:            Language
  Type:            REG_DWORD
  Data:            0x409

Value 5
  Name:            RegisteredOwner
  Type:            REG_SZ
  Data:            SYSTEM

Value 6
  Name:            SerialNumber
  Type:            REG_DWORD
  Data:            0x81e40040


Key Name:          SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\Parameters
Class Name:        <NO CLASS>
Last Write Time:   6/16/99 - 3:26 PM
Value 0
  Name:            SQLArg0
  Type:            REG_SZ
  Data:            -dC:\MSSQL7\data\master.mdf

Value 1
  Name:            SQLArg1
  Type:            REG_SZ
  Data:            -eC:\MSSQL7\log\ERRORLOG

Value 2
```

```
  Name:            SQLArg2
  Type:            REG_SZ
  Data:            -lC:\MSSQL7\data\mastlog.ldf


Key Name:          SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\RPCNetLib
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 7:47 AM
Value 0
  Name:            Security
  Type:            REG_SZ
  Data:


Key Name:          SOFTWARE\Microsoft\MSSQLServer\Providers
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            AllowInProcess
  Type:            REG_DWORD
  Data:            0x1


Key Name:          SOFTWARE\Microsoft\MSSQLServer\Providers\ADSDSOObject
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            AllowInProcess
  Type:            REG_DWORD
  Data:            0x1


Key Name:          SOFTWARE\Microsoft\MSSQLServer\Providers\DTSPackageDSO
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            AllowInProcess
  Type:            REG_DWORD
  Data:            0x1


Key Name:
SOFTWARE\Microsoft\MSSQLServer\Providers\Microsoft.Jet.OLEDB.4.0
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            AllowInProcess
  Type:            REG_DWORD
  Data:            0x1


Key Name:          SOFTWARE\Microsoft\MSSQLServer\Providers\MSDAORA
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            AllowInProcess
  Type:            REG_DWORD
  Data:            0x1


Key Name:          SOFTWARE\Microsoft\MSSQLServer\Providers\MSDASQL
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            AllowInProcess
  Type:            REG_DWORD
  Data:            0x1


Key Name:          SOFTWARE\Microsoft\MSSQLServer\Providers\MSIDXS
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            AllowInProcess
  Type:            REG_DWORD
  Data:            0x1


Key Name:          SOFTWARE\Microsoft\MSSQLServer\Providers\MSOLAP
Class Name:        <NO CLASS>
Last Write Time:   6/16/99 - 4:23 PM
Value 0
  Name:            AllowInProcess
  Type:            REG_DWORD
  Data:            0x1


Key Name:          SOFTWARE\Microsoft\MSSQLServer\Providers\MSQLImpProv
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            AllowInProcess
  Type:            REG_DWORD
  Data:            0x1


Key Name:          SOFTWARE\Microsoft\MSSQLServer\Providers\MSSEARCHSQL
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            AllowInProcess
  Type:            REG_DWORD
  Data:            0x1


Key Name:          SOFTWARE\Microsoft\MSSQLServer\Providers\SQLOLEDB
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            AllowInProcess
  Type:            REG_DWORD
  Data:            0x1


Key Name:          SOFTWARE\Microsoft\MSSQLServer\Replication
Class Name:        <NO CLASS>
Last Write Time:   6/10/98 - 1:00 PM
```

```
Key Name:
SOFTWARE\Microsoft\MSSQLServer\Replication\MergeReplicationProvider
Class Name:         <NO CLASS>
Last Write Time:    6/10/98 - 1:00 PM

Key Name:
SOFTWARE\Microsoft\MSSQLServer\Replication\MergeReplicationProvider\7.0
Class Name:         <NO CLASS>
Last Write Time:    6/10/98 - 1:00 PM

Key Name:
SOFTWARE\Microsoft\MSSQLServer\Replication\MergeReplicationProvider\7.0\Ms
Jet
Class Name:         <NO CLASS>
Last Write Time:    6/16/99 - 4:23 PM
Value 0
  Name:             <NO NAME>
  Type:             REG_SZ
  Data:             {f159cf30-0db4-11d1-b272-00aa00b8de95}


Key Name:           SOFTWARE\Microsoft\MSSQLServer\Setup
Class Name:         <NO CLASS>
Last Write Time:    6/11/99 - 7:47 AM
Value 0
  Name:             SourcePath
  Type:             REG_SZ
  Data:             Z:\Sql70SP2Beta

Value 1
  Name:             SQLDataRoot
  Type:             REG_SZ
  Data:             C:\MSSQL7

Value 2
  Name:             SQLPath
  Type:             REG_SZ
  Data:             C:\MSSQL7


Key Name:           SOFTWARE\Microsoft\MSSQLServer\SNMP
Class Name:         <NO CLASS>
Last Write Time:    11/13/98 - 10:27 AM

Key Name:           SOFTWARE\Microsoft\MSSQLServer\SNMP\CurrentVersion
Class Name:         <NO CLASS>
Last Write Time:    6/11/99 - 8:09 AM
Value 0
  Name:             Pathname
  Type:             REG_EXPAND_SZ
  Data:             C:\MSSQL7\BINN\sqlsnmp.dll


Key Name:           SOFTWARE\Microsoft\MSSQLServer\SQL Service Manager
Class Name:         <NO CLASS>
Last Write Time:    6/11/99 - 8:09 AM
Value 0
  Name:             Action Verify
  Type:             REG_DWORD
  Data:             0

Value 1
  Name:             DefaultSvc
  Type:             REG_SZ
  Data:             MSSQLServer

Value 2
  Name:             Remote
  Type:             REG_DWORD
  Data:             0x1

Value 3
  Name:             Services
  Type:             REG_MULTI_SZ
  Data:             MSSQLServer
                    SQLServerAgent
                    MSDTC


Key Name:           SOFTWARE\Microsoft\MSSQLServer\SQLEW
Class Name:         <NO CLASS>
Last Write Time:    1/11/99 - 12:31 PM

Key Name:           SOFTWARE\Microsoft\MSSQLServer\SQLEW\Replication
Class Name:         <NO CLASS>
Last Write Time:    6/11/99 - 8:09 AM
Value 0
  Name:             PerfmonFile
  Type:             REG_SZ
  Data:             C:\MSSQL7\BINN\REPLMON.PMC


Key Name:           SOFTWARE\Microsoft\MSSQLServer\SQLEW\Wizards
Class Name:         <NO CLASS>
Last Write Time:    6/11/99 - 8:09 AM
Value 0
  Name:             Web Assistant
  Type:             REG_SZ
  Data:             C:\MSSQL7\BINN\semwebwz.DLL^WebWizardEntry


Key Name:           SOFTWARE\Microsoft\MSSQLServer\SQLServerAgent
Class Name:         <NO CLASS>
Last Write Time:    6/11/99 - 8:09 AM
Value 0
  Name:             AlertForwardingSeverity
  Type:             REG_DWORD
  Data:             0x1

Value 1
  Name:             DownloadedMaxRows
  Type:             REG_DWORD
  Data:             0x64

Value 2
  Name:             ErrorLogFile
  Type:             REG_SZ
```

```
  Data:            C:\MSSQL7\LOG\SQLAGENT.OUT

Value 3
  Name:            ErrorLoggingLevel
  Type:            REG_DWORD
  Data:            0x3

Value 4
  Name:            JobHistoryMaxRows
  Type:            REG_DWORD
  Data:            0x3e8

Value 5
  Name:            JobHistoryMaxRowsPerJob
  Type:            REG_DWORD
  Data:            0x64

Value 6
  Name:            MSXServerName
  Type:            REG_SZ
  Data:

Value 7
  Name:            NonAlertableErrors
  Type:            REG_SZ
  Data:            1204,4002

Value 8
  Name:            RestartSQLServer
  Type:            REG_DWORD
  Data:            0x1

Value 9
  Name:            ServerHost
  Type:            REG_SZ
  Data:

Value 10
  Name:            WorkingDirectory
  Type:            REG_SZ
  Data:            C:\MSSQL7\JOBS


Key Name:
SOFTWARE\Microsoft\MSSQLServer\SQLServerAgent\Subsystems
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            ActiveScripting
  Type:            REG_SZ
  Data:
C:\MSSQL7\BINN\SQLATXSS.DLL,NULL,ActiveScriptStart,ActiveScriptEvent,Activ
eScriptStop,10

Value 1
  Name:            CmdExec
  Type:            REG_SZ
  Data:
C:\MSSQL7\BINN\SQLCMDSS.DLL,NULL,CmdExecStart,CmdEvent,CmdExecStop,10

Value 2
  Name:            Distribution
  Type:            REG_SZ
  Data:
C:\MSSQL7\BINN\SQLREPSS.DLL,C:\MSSQL7\BINN\DISTRIB.EXE,ReplStart,ReplEvent
,ReplStop,100

Value 3
  Name:            LogReader
  Type:            REG_SZ
  Data:
C:\MSSQL7\BINN\SQLREPSS.DLL,C:\MSSQL7\BINN\LOGREAD.EXE,ReplStart,ReplEvent
,ReplStop,25

Value 4
  Name:            Merge
  Type:            REG_SZ
  Data:
C:\MSSQL7\BINN\SQLREPSS.DLL,C:\MSSQL7\BINN\REPLMERG.EXE,ReplStart,ReplEven
t,ReplStop,100

Value 5
  Name:            Snapshot
  Type:            REG_SZ
  Data:
C:\MSSQL7\BINN\SQLREPSS.DLL,C:\MSSQL7\BINN\SNAPSHOT.EXE,ReplStart,ReplEven
t,ReplStop,100


Key Name:          SOFTWARE\Microsoft\MSSQLServer\Tracking
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            {E07FDDA4-5A21-11d2-9DAD-00C04F79D434}
  Type:            REG_SZ
  Data:

Value 1
  Name:            {E07FDDA8-5A21-11d2-9DAD-00C04F79D434}
  Type:            REG_SZ
  Data:

Value 2
  Name:            {E07FDDA9-5A21-11d2-9DAD-00C04F79D434}
  Type:            REG_SZ
  Data:

Value 3
  Name:            {E07FDDAA-5A21-11d2-9DAD-00C04F79D434}
  Type:            REG_SZ
  Data:

Value 4
  Name:            {E07FDDAC-5A21-11d2-9DAD-00C04F79D434}
  Type:            REG_SZ
  Data:

Value 5
```

```
  Name:            {E07FDDAD-5A21-11d2-9DAD-00C04F79D434}
  Type:            REG_SZ
  Data:

Value 6
  Name:            {E07FDDAE-5A21-11d2-9DAD-00C04F79D434}
  Type:            REG_SZ
  Data:

Value 7
  Name:            {E07FDDAF-5A21-11d2-9DAD-00C04F79D434}
  Type:            REG_SZ
  Data:

Value 8
  Name:            {E07FDDBE-5A21-11d2-9DAD-00C04F79D434}
  Type:            REG_SZ
  Data:

Value 9
  Name:            {E07FDDBF-5A21-11d2-9DAD-00C04F79D434}
  Type:            REG_SZ
  Data:

Value 10
  Name:            {E07FDDC0-5A21-11d2-9DAD-00C04F79D434}
  Type:            REG_SZ
  Data:

Value 11
  Name:            {E07FDDC8-5A21-11d2-9DAD-00C04F79D434}
  Type:            REG_SZ
  Data:
```

Software\Intel\E100B

```
Key Name:          SOFTWARE\Intel\E100B
Class Name:        <NO CLASS>
Last Write Time:   2/12/99 - 3:14 PM

Key Name:          SOFTWARE\Intel\E100B\CurrentVersion
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            Description
  Type:            REG_SZ
  Data:            Intel(R) PRO NDIS Driver

Value 1
  Name:            InstallDate
  Type:            REG_DWORD
  Data:            0x36e0128d

Value 2
  Name:            MajorVersion
  Type:            REG_DWORD
  Data:            0x30003

Value 3
  Name:            MinorVersion
  Type:            REG_DWORD
  Data:            0

Value 4
  Name:            OperationsSupport
  Type:            REG_DWORD
  Data:            0xff

Value 5
  Name:            RefCount
  Type:            REG_DWORD
  Data:            0x1

Value 6
  Name:            Review
  Type:            REG_DWORD
  Data:            0

Value 7
  Name:            ServiceName
  Type:            REG_SZ
  Data:            E100B

Value 8
  Name:            SoftwareType
  Type:            REG_SZ
  Data:            driver

Value 9
  Name:            Title
  Type:            REG_SZ
  Data:            Intel(R) PRO NDIS Driver


Key Name:          SOFTWARE\Intel\E100B\CurrentVersion\NetRules
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            bindable
  Type:            REG_MULTI_SZ
  Data:            E100BDriver E100BAdapter non exclusive 100


Value 1
  Name:            bindform
  Type:            REG_SZ
  Data:            "E100BSys" yes no container

Value 2
  Name:            class
  Type:            REG_MULTI_SZ
  Data:            E100BDriver basic


Value 3
```

```
  Name:            InfName
  Type:            REG_SZ
  Data:            oemnad18.inf

Value 4
  Name:            InfOption
  Type:            REG_SZ
  Data:            E100B

Value 5
  Name:            type
  Type:            REG_SZ
  Data:            E100BSys ndisDriver E100BDriver

Value 6
  Name:            use
  Type:            REG_SZ
  Data:            driver
```

Control\Session Manager\I/O System

```
Key Name:          SYSTEM\CurrentControlSet\Control\Session Manager\I/O
System
Class Name:        <NO CLASS>
Last Write Time:   1/28/99 - 10:44 AM
Value 0
  Name:            LargeIrpStackLocations
  Type:            REG_DWORD
  Data:            0x6
```

Services\E100B

```
Key Name:          SYSTEM\CurrentControlSet\Services\E100B
Class Name:        <NO CLASS>
Last Write Time:   6/18/99 - 10:28 AM
Value 0
  Name:            DisplayName
  Type:            REG_SZ
  Data:            Intel(R) PRO NDIS Driver

Value 1
  Name:            ErrorControl
  Type:            REG_DWORD
  Data:            0x1

Value 2
  Name:            Group
  Type:            REG_SZ
  Data:            NDIS

Value 3
  Name:            ImagePath
  Type:            REG_EXPAND_SZ
  Data:            \SystemRoot\System32\drivers\E100BNT.SYS

Value 4
  Name:            RequestedSystemResources
  Type:            REG_RESOURCE_REQUIREMENTS_LIST
  Data:
                   Interface Type:      Internal
                   Bus Number:          0
                   Slot Number:         0
                   List 0
                     Descriptor 0
                       Resource:        Interrupt
                       Option:          0x00000000
                       Disposition:     Shared
                       Type:            Level Sensitive
                       Minimum Vector:  0x10
                       Maximum Vector:  0x10

                     Descriptor 1
                       Resource:        Memory
                       Option:          0x00000001
                       Disposition:     Device Exclusive
                       Type:            Write Only
                       Length:          0x1000
                       Alignment:       0x1000
                       Minimum Address: 0xe3200000
                       Maximum Address: 0xe3200fff

                     Descriptor 2
                       Resource:        Memory
                       Option:          0x00000009
                       Disposition:     Device Exclusive
                       Type:            Write Only
                       Length:          0x1000
                       Alignment:       0x1000
                       Minimum Address: 0xe3200000
                       Maximum Address: 0xe3200fff

                     Descriptor 3
                       Resource:        Memory
                       Option:          0x00000008
                       Disposition:     Device Exclusive
                       Type:            Write Only
                       Length:          0x1000
                       Alignment:       0x1000
                       Minimum Address: 0xe2100000
                       Maximum Address: 0xe21fffff

                     Descriptor 4
                       Resource:        Port
                       Option:          0x00000001
                       Disposition:     Device Exclusive
                       Type:            Port
                       Length:          0x20
                       Alignment:       0x20
                       Minimum Address: 0x00001800
                       Maximum Address: 0x0000181f

                     Descriptor 5
                       Resource:        Port
                       Option:          0x00000008
```

```
    Disposition:     Device Exclusive
    Type:            Port
    Length:          0x20
    Alignment:       0x20
    Minimum Address: 0x00001800
    Maximum Address: 0x0000181f

  Descriptor 6
    Resource:        Memory
    Option:          0x00000001
    Disposition:     Device Exclusive
    Type:            Read / Write
    Length:          0x100000
    Alignment:       0x100000
    Minimum Address: 0xe2100000
    Maximum Address: 0xe21fffff

  Descriptor 7
    Resource:        Memory
    Option:          0x00000008
    Disposition:     Device Exclusive
    Type:            Read / Write
    Length:          0x100000
    Alignment:       0x100000
    Minimum Address: 0xe2100000
    Maximum Address: 0xe21fffff
```

```
Value 5
  Name:            Start
  Type:            REG_DWORD
  Data:            0x2

Value 6
  Name:            Type
  Type:            REG_DWORD
  Data:            0x1


Key Name:          SYSTEM\CurrentControlSet\Services\E100B\Enum
Class Name:        <NO CLASS>
Last Write Time:   6/18/99 - 10:26 AM
Value 0
  Name:            0
  Type:            REG_SZ
  Data:            Root\LEGACY_E100B\0000

Value 1
  Name:            Count
  Type:            REG_DWORD
  Data:            0x1

Value 2
  Name:            NextInstance
  Type:            REG_DWORD
  Data:            0x1
```

```
Key Name:          SYSTEM\CurrentControlSet\Services\E100B\Linkage
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:25 AM
Value 0
  Name:            Bind
  Type:            REG_MULTI_SZ
  Data:            \Device\E100B1


Value 1
  Name:            Export
  Type:            REG_MULTI_SZ
  Data:            \Device\E100B1


Value 2
  Name:            Route
  Type:            REG_MULTI_SZ
  Data:            "E100B1"



Key Name:
SYSTEM\CurrentControlSet\Services\E100B\Linkage\Disabled
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:25 AM
Value 0
  Name:            Bind
  Type:            REG_MULTI_SZ
  Data:

Value 1
  Name:            Export
  Type:            REG_MULTI_SZ
  Data:

Value 2
  Name:            Route
  Type:            REG_MULTI_SZ
  Data:


Key Name:          SYSTEM\CurrentControlSet\Services\E100B\Parameters
Class Name:        <NO CLASS>
Last Write Time:   2/12/99 - 3:14 PM

Key Name:          SYSTEM\CurrentControlSet\Services\E100B\Security
Class Name:        <NO CLASS>
Last Write Time:   2/12/99 - 3:14 PM
Value 0
  Name:            Security
  Type:            REG_BINARY
  Data:
00000000   01 00 14 80 c0 00 00 00 - cc 00 00 00 14 00 00 00
................
00000010   34 00 00 00 02 00 20 00 - 01 00 00 00 02 80 18 00  4.....
........
00000020   ff 01 0f 00 01 01 00 00 - 00 00 00 01 00 00 00 00
................
```

```
00000030   20 02 00 00 02 00 8c 00 - 05 00 00 00 00 00 18 00
................
00000040   8d 01 02 00 01 01 00 00 - 00 00 00 01 00 00 00 00
................
00000050   4f 00 4b 00 00 00 1c 00 - fd 01 02 00 01 02 00 00
O.K.............
00000060   00 00 00 05 20 00 00 00 - 23 02 00 00 44 00 44 00  ....
...#...D.D.
00000070   00 00 1c 00 ff 01 0f 00 - 01 02 00 00 00 00 00 05
................
00000080   20 00 00 00 20 02 00 00 - 44 00 44 00 00 00 1c 00   ...
...D.D.....
00000090   ff 01 0f 00 01 02 00 00 - 00 00 00 05 20 00 00 00  ............
...
000000a0   25 02 00 00 44 00 44 00 - 00 00 18 00 fd 01 02 00
%...D.D.........
000000b0   01 01 00 00 00 00 00 05 - 12 00 00 00 25 02 00 00
............%...
000000c0   01 01 00 00 00 00 00 05 - 12 00 00 00 01 01 00 00
................
000000d0   00 00 00 05 12 00 00 00 -                          ........
```

Services\E100B1

```
Key Name:          SYSTEM\CurrentControlSet\Services\E100B1
Class Name:        <NO CLASS>
Last Write Time:   2/12/99 - 3:14 PM
Value 0
  Name:            ErrorControl
  Type:            REG_DWORD
  Data:            0x1

Value 1
  Name:            Start
  Type:            REG_DWORD
  Data:            0x3

Value 2
  Name:            type
  Type:            REG_DWORD
  Data:            0x4


Key Name:          SYSTEM\CurrentControlSet\Services\E100B1\Linkage
Class Name:        <NO CLASS>
Last Write Time:   6/18/99 - 10:28 AM
Value 0
  Name:            Bind
  Type:            REG_MULTI_SZ
  Data:            \Device\E100B1


Value 1
  Name:            Export
  Type:            REG_MULTI_SZ
  Data:            \Device\E100B1


Value 2
  Name:            Route
  Type:            REG_MULTI_SZ
  Data:            "E100B1"


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\Linkage\Disabled
Class Name:        <NO CLASS>
Last Write Time:   2/12/99 - 3:14 PM

Key Name:          SYSTEM\CurrentControlSet\Services\E100B1\Parameters
Class Name:        <NO CLASS>
Last Write Time:   5/17/99 - 2:35 PM
Value 0
  Name:            Adaptive_IFS
  Type:            REG_DWORD
  Data:            0x1

Value 1
  Name:            BoardHasBridge
  Type:            REG_DWORD
  Data:            0

Value 2
  Name:            BusNumber
  Type:            REG_DWORD
  Data:            0

Value 3
  Name:            BusType
  Type:            REG_DWORD
  Data:            0x5

Value 4
  Name:            BusTypeLocal
  Type:            REG_DWORD
  Data:            0x5

Value 5
  Name:            Coalesce
  Type:            REG_DWORD
  Data:            0

Value 6
  Name:            CPUSaver
  Type:            REG_DWORD
  Data:            0xa00

Value 7
  Name:            ForceDpx
  Type:            REG_DWORD
  Data:            0

Value 8
  Name:            Location
  Type:            REG_SZ
```

```
  Data:           2000

Value 9
  Name:           MediaType
  Type:           REG_DWORD
  Data:           0x1

Value 10
  Name:           MWIEnable
  Type:           REG_DWORD
  Data:           0

Value 11
  Name:           NetworkAddress
  Type:           REG_SZ
  Data:

Value 12
  Name:           NumCoalesce
  Type:           REG_DWORD
  Data:           0x20

Value 13
  Name:           NumRfd
  Type:           REG_DWORD
  Data:           0x60

Value 14
  Name:           NumTbdPerTcb
  Type:           REG_DWORD
  Data:           0xc

Value 15
  Name:           NumTcb
  Type:           REG_DWORD
  Data:           0x40

Value 16
  Name:           PacketTagging
  Type:           REG_DWORD
  Data:           0

Value 17
  Name:           PcNic
  Type:           REG_DWORD
  Data:           0x1

Value 18
  Name:           RxDmaCount
  Type:           REG_DWORD
  Data:           0

Value 19
  Name:           RxFifo
  Type:           REG_DWORD
  Data:           0x8

Value 20
  Name:           SlotNumber
  Type:           REG_DWORD
  Data:           0x4

Value 21
  Name:           Speed
  Type:           REG_DWORD
  Data:           0

Value 22
  Name:           Threshold
  Type:           REG_DWORD
  Data:           0xc8

Value 23
  Name:           TxDmaCount
  Type:           REG_DWORD
  Data:           0

Value 24
  Name:           TxFifo
  Type:           REG_DWORD
  Data:           0x8

Value 25
  Name:           Txmitwait
  Type:           REG_DWORD
  Data:           0x1

Value 26
  Name:           UcodeSW
  Type:           REG_DWORD
  Data:           0x1

Value 27
  Name:           UnderrunRetry
  Type:           REG_DWORD
  Data:           0x1

Value 28
  Name:           UseIo
  Type:           REG_DWORD
  Data:           0x2

Value 29
  Name:           UseManualPCIAssign
  Type:           REG_DWORD
  Data:           0

Value 30
  Name:           VLanMode
  Type:           REG_DWORD
  Data:           0


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\Parameters\Tcpip
Class Name:       GenericClass
Last Write Time:  6/18/99 - 10:28 AM
Value 0
```

```
  Name:            DefaultGateway
  Type:            REG_MULTI_SZ
  Data:


Value 1
  Name:            EnableDHCP
  Type:            REG_DWORD
  Data:            0

Value 2
  Name:            IPAddress
  Type:            REG_MULTI_SZ
  Data:            192.168.91.214


Value 3
  Name:            IPInterfaceContext
  Type:            REG_DWORD
  Data:            0x1

Value 4
  Name:            IPInterfaceContextMax
  Type:            REG_DWORD
  Data:            0x1

Value 5
  Name:            LLInterface
  Type:            REG_SZ
  Data:

Value 6
  Name:            PPTPFiltering
  Type:            REG_DWORD
  Data:            0

Value 7
  Name:            RawIPAllowedProtocols
  Type:            REG_MULTI_SZ
  Data:            0


Value 8
  Name:            SubnetMask
  Type:            REG_MULTI_SZ
  Data:            255.255.255.0


Value 9
  Name:            TCPAllowedPorts
  Type:            REG_MULTI_SZ
  Data:            0


Value 10
  Name:            UDPAllowedPorts
  Type:            REG_MULTI_SZ
  Data:            0


Value 11
  Name:            UseZeroBroadcast
  Type:            REG_DWORD
  Data:            0


Key Name:          SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi
Class Name:        <NO CLASS>
Last Write Time:   2/12/99 - 3:14 PM

Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params
Class Name:        <NO CLASS>
Last Write Time:   2/12/99 - 3:14 PM

Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\Adaptive_IFS
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            Base
  Type:            REG_SZ
  Data:            10

Value 1
  Name:            Default
  Type:            REG_SZ
  Data:            1

Value 2
  Name:            Max
  Type:            REG_SZ
  Data:            255

Value 3
  Name:            Min
  Type:            REG_SZ
  Data:            0

Value 4
  Name:            MiniHelp
  Type:            REG_SZ
  Data:

Value 5
  Name:            ParamDesc
  Type:            REG_SZ
  Data:            Adaptive Inter-Frame Spacing

Value 6
  Name:            Scale
  Type:            REG_SZ
  Data:            1

Value 7
  Name:            Step
  Type:            REG_SZ
  Data:            1
```

```
Value 8
  Name:            Type
  Type:            REG_SZ
  Data:            int


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\Coalesce
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            Default
  Type:            REG_SZ
  Data:            0

Value 1
  Name:            MiniHelp
  Type:            REG_SZ
  Data:

Value 2
  Name:            ParamDesc
  Type:            REG_SZ
  Data:            PCI Bus Efficiency

Value 3
  Name:            Type
  Type:            REG_SZ
  Data:            enum


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\Coalesce\Enum
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            0
  Type:            REG_SZ
  Data:            Disabled

Value 1
  Name:            1
  Type:            REG_SZ
  Data:            Enabled


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\CPUSaver
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            Default
  Type:            REG_SZ
  Data:            1536

Value 1
  Name:            LeftLabel
  Type:            REG_SZ
  Data:            Adapter Bandwidth

Value 2
  Name:            MiniHelp
  Type:            REG_SZ
  Data:            Sets optimal point of CPU/Adapter performance for this
system.  See help.

Value 3
  Name:            ParamDesc
  Type:            REG_SZ
  Data:            Adaptive Performance Tuning

Value 4
  Name:            RightLabel
  Type:            REG_SZ
  Data:            CPU Utilization

Value 5
  Name:            Type
  Type:            REG_SZ
  Data:            slider


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\CPUSaver\Values
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            0
  Type:            REG_SZ
  Data:            0

Value 1
  Name:            1
  Type:            REG_SZ
  Data:            256

Value 2
  Name:            10
  Type:            REG_SZ
  Data:            2560

Value 3
  Name:            11
  Type:            REG_SZ
  Data:            2816

Value 4
  Name:            12
  Type:            REG_SZ
  Data:            3072

Value 5
  Name:            13
  Type:            REG_SZ
  Data:            3328

Value 6
```

```
  Name:            14
  Type:            REG_SZ
  Data:            3584

Value 7
  Name:            15
  Type:            REG_SZ
  Data:            3840

Value 8
  Name:            16
  Type:            REG_SZ
  Data:            4096

Value 9
  Name:            2
  Type:            REG_SZ
  Data:            512

Value 10
  Name:            3
  Type:            REG_SZ
  Data:            768

Value 11
  Name:            4
  Type:            REG_SZ
  Data:            1024

Value 12
  Name:            5
  Type:            REG_SZ
  Data:            1280

Value 13
  Name:            6
  Type:            REG_SZ
  Data:            1536

Value 14
  Name:            7
  Type:            REG_SZ
  Data:            1792

Value 15
  Name:            8
  Type:            REG_SZ
  Data:            2048

Value 16
  Name:            9
  Type:            REG_SZ
  Data:            2304


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\ForceDpx
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            Default
  Type:            REG_SZ
  Data:            0

Value 1
  Name:            MiniHelp
  Type:            REG_SZ
  Data:

Value 2
  Name:            ParamDesc
  Type:            REG_SZ
  Data:            Duplex

Value 3
  Name:            Type
  Type:            REG_SZ
  Data:            enum


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\ForceDpx\Enum
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            0
  Type:            REG_SZ
  Data:            Auto Detect

Value 1
  Name:            1
  Type:            REG_SZ
  Data:            Half-Duplex

Value 2
  Name:            2
  Type:            REG_SZ
  Data:            Full-Duplex


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\NetworkAddress
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            Default
  Type:            REG_SZ
  Data:

Value 1
  Name:            MiniHelp
  Type:            REG_SZ
  Data:

Value 2
  Name:            ParamDesc
  Type:            REG_SZ
  Data:            Locally Administered Address
```

```
Value 3
  Name:            Type
  Type:            REG_SZ
  Data:            edit


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\NumCoalesce
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            Base
  Type:            REG_SZ
  Data:            10

Value 1
  Name:            Default
  Type:            REG_SZ
  Data:            8

Value 2
  Name:            Max
  Type:            REG_SZ
  Data:            32

Value 3
  Name:            Min
  Type:            REG_SZ
  Data:            1

Value 4
  Name:            MiniHelp
  Type:            REG_SZ
  Data:

Value 5
  Name:            ParamDesc
  Type:            REG_SZ
  Data:            Coalesce Buffers

Value 6
  Name:            Scale
  Type:            REG_SZ
  Data:            1

Value 7
  Name:            Step
  Type:            REG_SZ
  Data:            1

Value 8
  Name:            Type
  Type:            REG_SZ
  Data:            int


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\NumRfd
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            Base
  Type:            REG_SZ
  Data:            10

Value 1
  Name:            Default
  Type:            REG_SZ
  Data:            48

Value 2
  Name:            Max
  Type:            REG_SZ
  Data:            1024

Value 3
  Name:            Min
  Type:            REG_SZ
  Data:            1

Value 4
  Name:            MiniHelp
  Type:            REG_SZ
  Data:

Value 5
  Name:            ParamDesc
  Type:            REG_SZ
  Data:            Receive Buffers

Value 6
  Name:            Scale
  Type:            REG_SZ
  Data:            1

Value 7
  Name:            Step
  Type:            REG_SZ
  Data:            1

Value 8
  Name:            Type
  Type:            REG_SZ
  Data:            int


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\NumTcb
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            Base
  Type:            REG_SZ
  Data:            10

Value 1
  Name:            Default
```

```
  Type:            REG_SZ
  Data:            32

Value 2
  Name:            Max
  Type:            REG_SZ
  Data:            80

Value 3
  Name:            Min
  Type:            REG_SZ
  Data:            1

Value 4
  Name:            MiniHelp
  Type:            REG_SZ
  Data:

Value 5
  Name:            ParamDesc
  Type:            REG_SZ
  Data:            Transmit Control Blocks

Value 6
  Name:            Scale
  Type:            REG_SZ
  Data:            1

Value 7
  Name:            Step
  Type:            REG_SZ
  Data:            1

Value 8
  Name:            Type
  Type:            REG_SZ
  Data:            int


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\PacketTagging
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            Default
  Type:            REG_SZ
  Data:            0

Value 1
  Name:            MiniHelp
  Type:            REG_SZ
  Data:

Value 2
  Name:            ParamDesc
  Type:            REG_SZ
  Data:            802.1p/802.1q Tagging

Value 3
  Name:            Type
  Type:            REG_SZ
  Data:            enum


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\PacketTagging\En
um
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            0
  Type:            REG_SZ
  Data:            Disabled

Value 1
  Name:            1
  Type:            REG_SZ
  Data:            Enabled


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\Speed
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            Default
  Type:            REG_SZ
  Data:            0

Value 1
  Name:            MiniHelp
  Type:            REG_SZ
  Data:

Value 2
  Name:            ParamDesc
  Type:            REG_SZ
  Data:            Speed

Value 3
  Name:            Type
  Type:            REG_SZ
  Data:            enum


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\Speed\Enum
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            0
  Type:            REG_SZ
  Data:            Auto Detect

Value 1
  Name:            10
  Type:            REG_SZ
  Data:            10 Mbps
```

```
Value 2
  Name:            100
  Type:            REG_SZ
  Data:            100 Mbps


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\Threshold
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            Base
  Type:            REG_SZ
  Data:            10

Value 1
  Name:            Default
  Type:            REG_SZ
  Data:            16

Value 2
  Name:            Max
  Type:            REG_SZ
  Data:            200

Value 3
  Name:            Min
  Type:            REG_SZ
  Data:            0

Value 4
  Name:            MiniHelp
  Type:            REG_SZ
  Data:

Value 5
  Name:            ParamDesc
  Type:            REG_SZ
  Data:            Adaptive Transmit Threshold

Value 6
  Name:            Scale
  Type:            REG_SZ
  Data:            1

Value 7
  Name:            Step
  Type:            REG_SZ
  Data:            1

Value 8
  Name:            Type
  Type:            REG_SZ
  Data:            int


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\UcodeSW
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            Default
  Type:            REG_SZ
  Data:            1

Value 1
  Name:            MiniHelp
  Type:            REG_SZ
  Data:

Value 2
  Name:            ParamDesc
  Type:            REG_SZ
  Data:            Adaptive Technology

Value 3
  Name:            Type
  Type:            REG_SZ
  Data:            enum


Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\UcodeSW\Enum
Class Name:        <NO CLASS>
Last Write Time:   3/5/99 - 10:21 AM
Value 0
  Name:            0
  Type:            REG_SZ
  Data:            Off

Value 1
  Name:            1
  Type:            REG_SZ
  Data:            On
```

Services\mraid

```
Key Name:          SYSTEM\CurrentControlSet\Services\mraid
Class Name:        REG_DWORD
Last Write Time:   9/15/99 - 3:46 PM
Value 0
  Name:            ErrorControl
  Type:            REG_DWORD
  Data:            0

Value 1
  Name:            Group
  Type:            REG_SZ
  Data:            Primary disk

Value 2
  Name:            Start
  Type:            REG_DWORD
  Data:            0
```

```
Value 3
  Name:            Tag
  Type:            REG_DWORD
  Data:            0x1

Value 4
  Name:            Type
  Type:            REG_DWORD
  Data:            0x1


Key Name:          SYSTEM\CurrentControlSet\Services\mraid\Enum
Class Name:        <NO CLASS>
Last Write Time:   10/4/99 - 3:04 PM
Value 0
  Name:            0
  Type:            REG_SZ
  Data:            Root\LEGACY_MRAID\0000

Value 1
  Name:            Count
  Type:            REG_DWORD
  Data:            0x1

Value 2
  Name:            NextInstance
  Type:            REG_DWORD
  Data:            0x1
```

Services\MSSQLServer

```
Key Name:          SYSTEM\CurrentControlSet\Services\MSSQLServer
Class Name:        <NO CLASS>
Last Write Time:   1/11/99 - 12:31 PM
Value 0
  Name:            DisplayName
  Type:            REG_SZ
  Data:            MSSQLServer

Value 1
  Name:            ErrorControl
  Type:            REG_DWORD
  Data:            0x1

Value 2
  Name:            ImagePath
  Type:            REG_EXPAND_SZ
  Data:            C:\MSSQL7\binn\sqlservr.exe

Value 3
  Name:            ObjectName
  Type:            REG_SZ
  Data:            LocalSystem

Value 4
  Name:            Start
  Type:            REG_DWORD
  Data:            0x3

Value 5
  Name:            Type
  Type:            REG_DWORD
  Data:            0x10


Key Name:          SYSTEM\CurrentControlSet\Services\MSSQLServer\Enum
Class Name:        <NO CLASS>
Last Write Time:   6/18/99 - 10:26 AM
Value 0
  Name:            0
  Type:            REG_SZ
  Data:            Root\LEGACY_MSSQLSERVER\0000

Value 1
  Name:            Count
  Type:            REG_DWORD
  Data:            0x1

Value 2
  Name:            NextInstance
  Type:            REG_DWORD
  Data:            0x1


Key Name:
SYSTEM\CurrentControlSet\Services\MSSQLServer\Performance
Class Name:        <NO CLASS>
Last Write Time:   6/11/99 - 8:09 AM
Value 0
  Name:            Close
  Type:            REG_SZ
  Data:            CloseSQLPerformanceData

Value 1
  Name:            Collect
  Type:            REG_SZ
  Data:            CollectSQLPerformanceData

Value 2
  Name:            First Counter
  Type:            REG_DWORD
  Data:            0x738

Value 3
  Name:            First Help
  Type:            REG_DWORD
  Data:            0x739

Value 4
  Name:            Last Counter
  Type:            REG_DWORD
  Data:            0x83a

Value 5
```

```
  Name:            Last Help
  Type:            REG_DWORD
  Data:            0x83b

Value 6
  Name:            Library
  Type:            REG_SZ
  Data:            SQLCTR70.DLL

Value 7
  Name:            Open
  Type:            REG_SZ
  Data:            OpenSQLPerformanceData


Key Name:          SYSTEM\CurrentControlSet\Services\MSSQLServer\Security
Class Name:        <NO CLASS>
Last Write Time:   1/11/99 - 12:28 PM
Value 0
  Name:            Security
  Type:            REG_BINARY
  Data:
00000000   01 00 14 80 c0 00 00 00 - cc 00 00 00 14 00 00 00
................
00000010   34 00 00 00 02 00 20 00 - 01 00 00 00 02 80 18 00  4.....
.........
00000020   ff 01 0f 00 01 01 00 00 - 00 00 00 01 00 00 00 00
................
00000030   20 02 00 00 02 00 8c 00 - 05 00 00 00 00 00 18 00
...............
00000040   8d 01 02 00 01 01 00 00 - 00 00 00 01 00 00 00 00
................
00000050   74 00 69 00 00 00 1c 00 - fd 01 02 00 01 02 00 00
t.i.............
00000060   00 00 00 05 20 00 00 00 - 23 02 00 00 6f 00 6e 00  ....
...#...o.n.
00000070   00 00 1c 00 ff 01 0f 00 - 01 02 00 00 00 00 00 05
................
00000080   20 00 00 00 20 02 00 00 - 6f 00 6e 00 00 00 1c 00   ...
...o.n.....
00000090   ff 01 0f 00 01 02 00 00 - 00 00 00 05 20 00 00 00  ............
...
000000a0   25 02 00 00 6f 00 6e 00 - 00 00 18 00 fd 01 02 00
%...o.n.........
000000b0   01 01 00 00 00 00 00 05 - 12 00 00 00 25 02 00 00
............%...
000000c0   01 01 00 00 00 00 00 05 - 12 00 00 00 01 01 00 00
................
000000d0   00 00 00 05 12 00 00 00 -                          ........
```

Services\NDIS

```
Key Name:          SYSTEM\CurrentControlSet\Services\NDIS
       Class Name:        <NO CLASS>
       Last Write Time:   10/10/96 - 12:09 AM
       Value 0
         Name:            DisplayName
         Type:            REG_SZ
         Data:            Microsoft NDIS System Driver

       Value 1
         Name:            ErrorControl
         Type:            REG_DWORD
         Data:            0x1

       Value 2
         Name:            Group
         Type:            REG_SZ
         Data:            NDIS

       Value 3
         Name:            Start
         Type:            REG_DWORD
         Data:            0x1

       Value 4
         Name:            Type
         Type:            REG_DWORD
         Data:            0x1


       Key Name:          SYSTEM\CurrentControlSet\Services\NDIS\Enum
       Class Name:        <NO CLASS>
       Last Write Time:   3/1/99 - 11:12 AM
       Value 0
         Name:            0
         Type:            REG_SZ
         Data:            Root\LEGACY_NDIS\0000

       Value 1
         Name:            Count
         Type:            REG_DWORD
         Data:            0x1

       Value 2
         Name:            NextInstance
         Type:            REG_DWORD
         Data:            0x1


       Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\MediaTypes
       Class Name:        <NO CLASS>
       Last Write Time:   10/10/96 - 12:09 AM

       Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\Parameters
       Class Name:        <NO CLASS>
       Last Write Time:   6/10/98 - 1:17 PM
       Value 0
         Name:            ProcessorAffinityMask
         Type:            REG_DWORD
         Data:            0
```

Services\NetBT

```
Key Name:          SYSTEM\CurrentControlSet\Services\NetBT
Class Name:        <NO CLASS>
Last Write Time:   6/10/98 - 4:07 AM
Value 0
  Name:            DependOnGroup
  Type:            REG_MULTI_SZ
  Data:

Value 1
  Name:            DependOnService
  Type:            REG_MULTI_SZ
  Data:            Tcpip


Value 2
  Name:            DisplayName
  Type:            REG_SZ
  Data:            WINS Client(TCP/IP)

Value 3
  Name:            ErrorControl
  Type:            REG_DWORD
  Data:            0x1

Value 4
  Name:            Group
  Type:            REG_SZ
  Data:            PNP_TDI

Value 5
  Name:            ImagePath
  Type:            REG_EXPAND_SZ
  Data:            \SystemRoot\System32\drivers\netbt.sys

Value 6
  Name:            Start
  Type:            REG_DWORD
  Data:            0x2

Value 7
  Name:            Type
  Type:            REG_DWORD
  Data:            0x1


Key Name:          SYSTEM\CurrentControlSet\Services\NetBT\Adapters
Class Name:        GenericClass
Last Write Time:   6/10/98 - 4:05 AM

Key Name:
SYSTEM\CurrentControlSet\Services\NetBT\Adapters\E1
                   00B1
Class Name:        GenericClass
Last Write Time:   6/10/98 - 4:07 AM
Value 0
  Name:            NameServer
  Type:            REG_SZ
  Data:

Value 1
  Name:            NameServerBackup
  Type:            REG_SZ
  Data:


Key Name:          SYSTEM\CurrentControlSet\Services\NetBT\Enum
Class Name:        <NO CLASS>
Last Write Time:   6/17/98 - 6:48 PM
Value 0
  Name:            Count
  Type:            REG_DWORD
  Data:            0

Value 1
  Name:            NextInstance
  Type:            REG_DWORD
  Data:            0


Key Name:          SYSTEM\CurrentControlSet\Services\NetBT\Linkage
Class Name:        GenericClass
Last Write Time:   6/10/98 - 4:07 AM
Value 0
  Name:            Bind
  Type:            REG_MULTI_SZ
  Data:

Value 1
  Name:            Export
  Type:            REG_MULTI_SZ
  Data:

Value 2
  Name:            OtherDependencies
  Type:            REG_MULTI_SZ
  Data:            Tcpip


Value 3
  Name:            Route
  Type:            REG_MULTI_SZ
  Data:


Key Name:
SYSTEM\CurrentControlSet\Services\NetBT\Linkage\Dis
                   abled
Class Name:        GenericClass
Last Write Time:   6/10/98 - 4:07 AM
Value 0
  Name:            Bind
  Type:            REG_MULTI_SZ
  Data:            \Device\E100B1
```

```
        Value 1
          Name:            Export
          Type:            REG_MULTI_SZ
          Data:            \Device\NetBT_E100B1


        Value 2
          Name:            Route
          Type:            REG_MULTI_SZ
          Data:            "E100B" "E100B1"



        Key Name:
SYSTEM\CurrentControlSet\Services\NetBT\Parameters
        Class Name:        GenericClass
        Last Write Time:   6/10/98 - 4:07 AM
        Value 0
          Name:            BcastNameQueryCount
          Type:            REG_DWORD
          Data:            0x3

        Value 1
          Name:            BcastQueryTimeout
          Type:            REG_DWORD
          Data:            0x2ee

        Value 2
          Name:            CacheTimeout
          Type:            REG_DWORD
          Data:            0x927c0

        Value 3
          Name:            EnableDNS
          Type:            REG_DWORD
          Data:            0

        Value 4
          Name:            EnableLMHOSTS
          Type:            REG_DWORD
          Data:            0x1

        Value 5
          Name:            EnableProxy
          Type:            REG_DWORD
          Data:            0

        Value 6
          Name:            NameServerPort
          Type:            REG_DWORD
          Data:            0x89

        Value 7
          Name:            NameSrvQueryCount
          Type:            REG_DWORD
          Data:            0x3

        Value 8
          Name:            NameSrvQueryTimeout
          Type:            REG_DWORD
          Data:            0x5dc

        Value 9
          Name:            NbProvider
          Type:            REG_SZ
          Data:            _tcp

        Value 10
          Name:            ScopeID
          Type:            REG_SZ
          Data:

        Value 11
          Name:            SessionKeepAlive
          Type:            REG_DWORD
          Data:            0x36ee80

        Value 12
          Name:            Size/Small/Medium/Large
          Type:            REG_DWORD
          Data:            0x1

        Value 13
          Name:            TransportBindName
          Type:            REG_SZ
          Data:            \Device\


        Key Name:          SYSTEM\CurrentControlSet\Services\NetBT\Security
        Class Name:        <NO CLASS>
        Last Write Time:   6/10/98 - 4:05 AM
        Value 0
          Name:            Security
          Type:            REG_BINARY
          Data:
        00000000   01 00 14 80 c0 00 00 00 - cc 00 00 00 14 00 00 00
........
                   ........
        00000010   34 00 00 00 02 00 20 00 - 01 00 00 00 02 80 18 00
4..... .
                   ........
        00000020   ff 01 0f 00 01 01 00 00 - 00 00 00 01 00 00 00 00
........
                   ........
        00000030   20 02 00 00 02 00 8c 00 - 05 00 00 00 00 00 18 00
.......
                   ........
        00000040   8d 01 02 00 01 01 00 00 - 00 00 00 01 00 00 00 00
........
                   ........
        00000050   01 01 00 00 00 00 1c 00 - fd 01 02 00 01 02 00 00
........
                   ........
        00000060   00 00 00 05 20 00 00 00 - 23 02 00 00 00 00 00 05  ....
...
                   #.......
        00000070   00 00 1c 00 ff 01 0f 00 - 01 02 00 00 00 00 00 05
........
```

```
       ........
       00000080   20 00 00 00 20 02 00 00 - 00 00 00 05 00 00 1c 00   ...
...
       ........
       00000090   ff 01 0f 00 01 02 00 00 - 00 00 00 05 20 00 00 00
........
       .... ...
       000000a0   25 02 00 00 00 00 00 05 - 00 00 18 00 fd 01 02 00
%.......
       ........
       000000b0   01 01 00 00 00 00 00 05 - 12 00 00 00 25 02 00 00
........
       ....%...
       000000c0   01 01 00 00 00 00 00 05 - 12 00 00 00 01 01 00 00
........
       ........
       000000d0   00 00 00 05 12 00 00 00 -
........
```

Services\PROSet

```
Key Name:          SYSTEM\CurrentControlSet\Services\PROSet
Class Name:        GenericClass
Last Write Time:   6/10/98 - 4:01 AM

Key Name:          SYSTEM\CurrentControlSet\Services\PROSet\Adapters
Class Name:        GenericClass
Last Write Time:   2/1/99 - 10:09 AM
Value 0
  Name:            EPRO100
  Type:            REG_SZ
  Data:            Intel EtherExpress PRO Adapter


Key Name:          SYSTEM\CurrentControlSet\Services\PROSet\EPRO100
Class Name:        GenericClass
Last Write Time:   2/1/99 - 10:09 AM
Value 0
  Name:            AdapterDescription
  Type:            REG_SZ
  Data:            EPRO100_GetAdapterDescription

Value 1
  Name:            Configure
  Type:            REG_SZ
  Data:            EPRO100_Configure

Value 2
  Name:            Detect
  Type:            REG_SZ
  Data:            EPRO100_Detect

Value 3
  Name:            DeviceExist
  Type:            REG_SZ
  Data:            EPRO100_DeviceExist

Value 4
  Name:            Diagnose
  Type:            REG_SZ
  Data:            EPRO100_Diagnose

Value 5
  Name:            DLL
  Type:            REG_SZ
  Data:            EPRO100.DLL

Value 6
  Name:            GetExtendedFeatures
  Type:            REG_SZ
  Data:            EPRO100_GetExtendedFeatures

Value 7
  Name:            Help
  Type:            REG_SZ
  Data:            E100SET.HLP

Value 8
  Name:            InstallAnyway
  Type:            REG_SZ
  Data:            EPRO100_InstallAnyway

Value 9
  Name:            RegistryKey
  Type:            REG_SZ
  Data:            EPRO100_GetRegistryKey

Value 10
  Name:            Summary
  Type:            REG_SZ
  Data:            EPRO100_Resource_Summary


Key Name:
SYSTEM\CurrentControlSet\Services\PROSet\EPRO100\Parameters
Class Name:        GenericClass
Last Write Time:   2/1/99 - 10:09 AM
Value 0
  Name:            Adaptive_IFS
  Type:            REG_SZ
  Data:            1,7,Adaptive Inter-Frame Spacing,0,2,1,0,255,1

Value 1
  Name:            BusNumber
  Type:            REG_SZ
  Data:            0,7,BusNumber,0,2,0,0,16,1

Value 2
  Name:            BusType
  Type:            REG_SZ
  Data:            0,7,BusType,0,2,5,2,5,1

Value 3
  Name:            BusTypeLocal
  Type:            REG_SZ
```

```
  Data:           0,7,BusTypeLocal,0,2,5,2,5,1

Value 4
  Name:           Eid
  Type:           REG_SZ
  Data:           0,7,Eid,0,2,0,0,4294967295,1

Value 5
  Name:           Fifo
  Type:           REG_SZ
  Data:           0,3,Fifo Depth,0,2,12,0,15,1

Value 6
  Name:           ForceDpx
  Type:           REG_SZ
  Data:           1,4,Duplex Mode,0,1,Auto,Auto,Half,Full

Value 7
  Name:           MapRegisters
  Type:           REG_SZ
  Data:

Value 8
  Name:           MediaType
  Type:           REG_SZ
  Data:           0,7,MediaType,0,2,1,1,1,1

Value 9
  Name:           MsPciScan
  Type:           REG_SZ
  Data:           0,4,MsPciScan,0,2,1,0,1,1

Value 10
  Name:           NetworkAddress
  Type:           REG_SZ
  Data:           1,7,Locally Administered Address,0,5,0,0,1,1

Value 11
  Name:           NumCoalesce
  Type:           REG_SZ
  Data:           1,7,Coalesce Buffers,0,2,8,1,32,1

Value 12
  Name:           NumRfd
  Type:           REG_SZ
  Data:           1,7,Receive Buffers,0,2,32,1,1024,1

Value 13
  Name:           NumTbd
  Type:           REG_SZ
  Data:           0,3,Transmit Buffer Descriptors,0,2,64,1,65535,1

Value 14
  Name:           NumTbdPerTcb
  Type:           REG_SZ
  Data:           0,4,Transmit Buffers per Frame,0,2,12,1,16,1

Value 15
  Name:           NumTcb
  Type:           REG_SZ
  Data:           1,7,Transmit Control Blocks,0,2,16,1,80,1

Value 16
  Name:           Off
  Type:           REG_SZ
  Data:           1,3,Off Timer,0,2,2,1,65535,1

Value 17
  Name:           On
  Type:           REG_SZ
  Data:           1,3,On Timer,0,2,32768,1,65535,1

Value 18
  Name:           PerfOptims
  Type:           REG_SZ
  Data:           0,4,PerfOptims,0,2,0,0,65535,1

Value 19
  Name:           RxDmaCount
  Type:           REG_SZ
  Data:           0,4,RxDmaCount,0,2,0,0,63,1

Value 20
  Name:           RxFifo
  Type:           REG_SZ
  Data:           0,4,Receive Fifo Depth,0,2,8,0,15,1

Value 21
  Name:           Slot
  Type:           REG_SZ
  Data:

Value 22
  Name:           Speed
  Type:           REG_SZ
  Data:           1,7,Network Speed,0,4,Auto,Auto,0,10Mbps,10,100Mbps,100

Value 23
  Name:           Threshold
  Type:           REG_SZ
  Data:           0,7,Transmit Threshold,0,2,16,0,200,1

Value 24
  Name:           TxDmaCount
  Type:           REG_SZ
  Data:           0,4,TxDmaCount,0,2,0,0,63,1

Value 25
  Name:           TxFifo
  Type:           REG_SZ
  Data:           0,4,Transmit Fifo Depth,0,2,8,0,15,1

Value 26
  Name:           Txmitwait
  Type:           REG_SZ
  Data:           0,7,Txmitwait,0,2,1,0,255,1

Value 27
```

```
  Name:            UcodeSW
  Type:            REG_SZ
  Data:            0,7,UcodeSW,0,2,1,0,1,1

Value 28
  Name:            UnderrunRetry
  Type:            REG_SZ
  Data:            0,4,UnderrunRetry,0,2,1,0,3,1
```

Services\Tcpip

```
Key Name:          SYSTEM\CurrentControlSet\Services\Tcpip
Class Name:        <NO CLASS>
Last Write Time:   6/10/98 - 4:06 AM
Value 0
  Name:            DisplayName
  Type:            REG_SZ
  Data:            TCP/IP Service

Value 1
  Name:            ErrorControl
  Type:            REG_DWORD
  Data:            0x1

Value 2
  Name:            Group
  Type:            REG_SZ
  Data:            PNP_TDI

Value 3
  Name:            ImagePath
  Type:            REG_EXPAND_SZ
  Data:            \SystemRoot\System32\drivers\tcpip.sys

Value 4
  Name:            Start
  Type:            REG_DWORD
  Data:            0x2

Value 5
  Name:            Type
  Type:            REG_DWORD
  Data:            0x1


Key Name:          SYSTEM\CurrentControlSet\Services\Tcpip\Enum
Class Name:        <NO CLASS>
Last Write Time:   6/18/99 - 10:26 AM
Value 0
  Name:            0
  Type:            REG_SZ
  Data:            Root\LEGACY_TCPIP\0000

Value 1
  Name:            Count
  Type:            REG_DWORD
  Data:            0x1

Value 2
  Name:            NextInstance
  Type:            REG_DWORD
  Data:            0x1


Key Name:          SYSTEM\CurrentControlSet\Services\Tcpip\Linkage
Class Name:        GenericClass
Last Write Time:   3/5/99 - 10:25 AM
Value 0
  Name:            Bind
  Type:            REG_MULTI_SZ
  Data:            \Device\E100B1


Value 1
  Name:            Export
  Type:            REG_MULTI_SZ
  Data:            \Device\Tcpip\E100B1


Value 2
  Name:            Route
  Type:            REG_MULTI_SZ
  Data:            "E100B" "E100B1"



Key Name:
SYSTEM\CurrentControlSet\Services\Tcpip\Linkage\Disabled
Class Name:        GenericClass
Last Write Time:   3/5/99 - 10:25 AM
Value 0
  Name:            Bind
  Type:            REG_MULTI_SZ
  Data:

Value 1
  Name:            Export
  Type:            REG_MULTI_SZ
  Data:

Value 2
  Name:            Route
  Type:            REG_MULTI_SZ
  Data:


Key Name:          SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
Class Name:        GenericClass
Last Write Time:   2/12/99 - 3:15 PM
Value 0
  Name:            DataBasePath
  Type:            REG_EXPAND_SZ
  Data:            %SystemRoot%\System32\drivers\etc

Value 1
  Name:            Domain
```

```
  Type:            REG_SZ
  Data:            mv.unisys.com

Value 2
  Name:            EnableSecurityFilters
  Type:            REG_DWORD
  Data:            0

Value 3
  Name:            ForwardBroadcasts
  Type:            REG_DWORD
  Data:            0

Value 4
  Name:            Hostname
  Type:            REG_SZ
  Data:            avalon4

Value 5
  Name:            IPEnableRouter
  Type:            REG_DWORD
  Data:            0

Value 6
  Name:            KeepAliveInterval
  Type:            REG_DWORD
  Data:            0x2710

Value 7
  Name:            KeepAliveTime
  Type:            REG_DWORD
  Data:            0x493e0

Value 8
  Name:            NameServer
  Type:            REG_SZ
  Data:

Value 9
  Name:            SearchList
  Type:            REG_SZ
  Data:

Value 10
  Name:            TcpAverageRTT
  Type:            REG_DWORD
  Data:            0x3e8


Key Name:
SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\PersistentRoutes
Class Name:        GenericClass
Last Write Time:   6/10/98 - 4:05 AM

Key Name:
SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Winsock
Class Name:        GenericClass
Last Write Time:   6/10/98 - 4:05 AM
Value 0
  Name:            HelperDllName
  Type:            REG_EXPAND_SZ
  Data:            %SystemRoot%\System32\wshtcpip.dll

Value 1
  Name:            Mapping
  Type:            REG_BINARY
  Data:
00000000   0b 00 00 00 03 00 00 00 - 02 00 00 00 01 00 00 00
................
00000010   06 00 00 00 02 00 00 00 - 01 00 00 00 00 00 00 00
................
00000020   02 00 00 00 00 00 00 00 - 06 00 00 00 00 00 00 00
................
00000030   00 00 00 00 06 00 00 00 - 00 00 00 00 01 00 00 00
................
00000040   06 00 00 00 02 00 00 00 - 02 00 00 00 11 00 00 00
................
00000050   02 00 00 00 02 00 00 00 - 00 00 00 00 02 00 00 00
................
00000060   00 00 00 00 11 00 00 00 - 00 00 00 00 00 00 00 00
................
00000070   11 00 00 00 00 00 00 00 - 02 00 00 00 11 00 00 00
................
00000080   02 00 00 00 03 00 00 00 - 00 00 00 00              ............

Value 2
  Name:            MaxSockAddrLength
  Type:            REG_DWORD
  Data:            0x10

Value 3
  Name:            MinSockAddrLength
  Type:            REG_DWORD
  Data:            0x10


Key Name:          SYSTEM\CurrentControlSet\Services\Tcpip\Performance
Class Name:        GenericClass
Last Write Time:   6/10/98 - 4:05 AM
Value 0
  Name:            Close
  Type:            REG_SZ
  Data:            CloseTcpIpPerformanceData

Value 1
  Name:            Collect
  Type:            REG_SZ
  Data:            CollectTcpIpPerformanceData

Value 2
  Name:            Library
  Type:            REG_SZ
  Data:            Perfctrs.dll

Value 3
  Name:            Open
  Type:            REG_SZ
  Data:            OpenTcpIpPerformanceData
```

```
Key Name:          SYSTEM\CurrentControlSet\Services\Tcpip\Security
Class Name:        <NO CLASS>
Last Write Time:   6/10/98 - 4:05 AM
Value 0
  Name:            Security
  Type:            REG_BINARY
  Data:
00000000   01 00 14 80 c0 00 00 00 - cc 00 00 00 14 00 00 00
................
00000010   34 00 00 00 02 00 20 00 - 01 00 00 00 02 80 18 00  4.....
........
00000020   ff 01 0f 00 01 01 00 00 - 00 00 00 01 00 00 00 00
................
00000030   20 02 00 00 02 00 8c 00 - 05 00 00 00 00 00 18 00
............
00000040   8d 01 02 00 01 01 00 00 - 00 00 00 01 00 00 00 00
................
00000050   6d 00 00 00 00 00 1c 00 - fd 01 02 00 01 02 00 00
m...............
00000060   00 00 00 05 20 00 00 00 - 23 02 00 00 43 00 48 00  ....
...#...C.H.
00000070   00 00 1c 00 ff 01 0f 00 - 01 02 00 00 00 00 00 05
................
00000080   20 00 00 00 20 02 00 00 - 43 00 48 00 00 00 1c 00   ...
...C.H.....
00000090   ff 01 0f 00 01 02 00 00 - 00 00 00 05 20 00 00 00  ............
...
000000a0   25 02 00 00 43 00 48 00 - 00 00 18 00 fd 01 02 00
%...C.H.........
000000b0   01 01 00 00 00 00 00 05 - 12 00 00 00 25 02 00 00
............%...
000000c0   01 01 00 00 00 00 00 05 - 12 00 00 00 01 01 00 00
................
000000d0   00 00 00 05 12 00 00 00 -                          ........


Key Name:          SYSTEM\CurrentControlSet\Services\Tcpip\ServiceProvider
Class Name:        GenericClass
Last Write Time:   6/10/98 - 4:05 AM
Value 0
  Name:            Class
  Type:            REG_DWORD
  Data:            0x8

Value 1
  Name:            DnsPriority
  Type:            REG_DWORD
  Data:            0x7d0

Value 2
  Name:            HostsPriority
  Type:            REG_DWORD
  Data:            0x1f4

Value 3
  Name:            LocalPriority
  Type:            REG_DWORD
  Data:            0x1f3

Value 4
  Name:            Name
  Type:            REG_SZ
  Data:            TCP/IP

Value 5
  Name:            NetbtPriority
  Type:            REG_DWORD
  Data:            0x7d1

Value 6
  Name:            ProviderPath
  Type:            REG_EXPAND_SZ
  Data:            %SystemRoot%\System32\wsock32.dll
```

Services\WinSock

```
      Key Name:          SYSTEM\CurrentControlSet\Services\WinSock
      Class Name:        GenericClass
      Last Write Time:   6/10/98 - 4:05 AM
      Value 0
        Name:            ErrorControl
        Type:            REG_DWORD
        Data:            0x1

      Value 1
        Name:            Start
        Type:            REG_DWORD
        Data:            0x3

      Value 2
        Name:            Type
        Type:            REG_DWORD
        Data:            0x4


      Key Name:
SYSTEM\CurrentControlSet\Services\WinSock\Autodial
      Class Name:        <NO CLASS>
      Last Write Time:   6/10/98 - 11:59 AM
      Value 0
        Name:            AutodialDllName32
        Type:            REG_SZ
        Data:            wininet.dll

      Value 1
        Name:            AutodialFcnName32
        Type:            REG_SZ
        Data:            InternetAutodialCallback


      Key Name:
SYSTEM\CurrentControlSet\Services\WinSock\Linkage
```

```
       Class Name:        GenericClass
       Last Write Time:   6/10/98 - 4:05 AM

       Key Name:
SYSTEM\CurrentControlSet\Services\WinSock\Linkage\D
                          isabled
       Class Name:        GenericClass
       Last Write Time:   6/10/98 - 4:05 AM

       Key Name:
SYSTEM\CurrentControlSet\Services\WinSock\Parameter
                          s
       Class Name:        GenericClass
       Last Write Time:   6/10/98 - 4:05 AM
       Value 0
         Name:            Transports
         Type:            REG_MULTI_SZ
         Data:            Tcpip
                          NetBIOS



       Key Name:          SYSTEM\CurrentControlSet\Services\WinSock\Setup
Mig
                          ration
       Class Name:        <NO CLASS>
       Last Write Time:   6/10/98 - 4:07 AM
       Value 0
         Name:            Known Static Providers
         Type:            REG_MULTI_SZ
         Data:            Tcpip
                          NwlnkIpx
                          NwlnkSpx
                          AppleTalk
                          IsoTp



       Value 1
         Name:            Provider List
         Type:            REG_MULTI_SZ
         Data:            Tcpip
                          NetBIOS



       Value 2
         Name:            Setup Version
         Type:            REG_DWORD
         Data:            0x1009



       Key Name:          SYSTEM\CurrentControlSet\Services\WinSock\Setup
Mig
                          ration\Providers
       Class Name:        <NO CLASS>
       Last Write Time:   6/10/98 - 4:06 AM

       Key Name:          SYSTEM\CurrentControlSet\Services\WinSock\Setup
Mig
                          ration\Providers\NetBIOS
       Class Name:        <NO CLASS>
       Last Write Time:   6/10/98 - 4:07 AM
       Value 0
         Name:            WinSock 1.1 Provider Data
         Type:            REG_BINARY
         Data:
       00000000   0e 10 00 00 11 00 00 00 - 14 00 00 00 14 00 00 00
........
       ........
       00000010   05 00 00 00 ff ff ff ff - 00 fa 00 00 66 00 00 00
........
       ....f...
       00000020   09 12 00 00 11 00 00 00 - 14 00 00 00 14 00 00 00
........
       ........
       00000030   02 00 00 00 ff ff ff ff - 00 fa 00 00 40 00 00 00
........
       ....@...
       00000040   5c 00 44 00 65 00 76 00 - 69 00 63 00 65 00 5c 00
\.D.e.v.
       i.c.e.\.
       00000050   4e 00 62 00 66 00 5f 00 - 45 00 31 00 30 00 30 00
N.b.f._.
       E.1.0.0.
       00000060   42 00 31 00 00 00 5c 00 - 44 00 65 00 76 00 69 00
B.1...\.
       D.e.v.i.
       00000070   63 00 65 00 5c 00 4e 00 - 62 00 66 00 5f 00 45 00
c.e.\.N.
       b.f._.E.
       00000080   31 00 30 00 30 00 42 00 - 31 00 00 00
1.0.0.B.
       1...

       Value 1
         Name:            WinSock 2.0 Provider ID
         Type:            REG_BINARY
         Data:
       00000000   30 18 5f 8d 73 c2 cf 11 - 95 c8 00 80 5f 48 a1 92
0._.s...
       ...._H..


       Key Name:          SYSTEM\CurrentControlSet\Services\WinSock\Setup
Mig
                          ration\Providers\Tcpip
       Class Name:        <NO CLASS>
       Last Write Time:   6/10/98 - 4:06 AM
       Value 0
         Name:            WinSock 2.0 Provider ID
         Type:            REG_BINARY
         Data:
       00000000   a0 1a 0f e7 8b ab cf 11 - 8c a3 00 80 5f 48 a1 92
........
       ...._H..


       Key Name:          SYSTEM\CurrentControlSet\Services\WinSock\Setup
Mig
                          ration\Well Known Guids
```

```
       Class Name:        <NO CLASS>
       Last Write Time:   6/10/98 - 4:06 AM
       Value 0
         Name:            AppleTalk
         Type:            REG_BINARY
         Data:
       00000000   a0 17 3b 2c df c6 cf 11 - 95 c8 00 80 5f 48 a1 92
..;,....
       ....._H..

       Value 1
         Name:            IsoTp
         Type:            REG_BINARY
         Data:
       00000000   b0 cb e4 89 c1 b9 cf 11 - 95 c8 00 80 5f 48 a1 92
........
       ....._H..

       Value 2
         Name:            McsXns
         Type:            REG_BINARY
         Data:
       00000000   b1 cb e4 89 c1 b9 cf 11 - 95 c8 00 80 5f 48 a1 92
........
       ....._H..
```

NT Client Configuration Information

```
System Information report written at: 10/21/99 05:06:32 PM
client1[System Information]

[ Following are sub-categories of this main category ]

[System Summary]

Item    Value
OS Name Microsoft Windows 2000 Server
Version 5.0.2072  Build 2072
OS Manufacturer        Microsoft Corporation
System Name    CLIENT1
System Manufacturer    HP
System Model   NetServer LC 3
System Type    X86-based PC
Processor      x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
BIOS Version
Windows Directory      C:\WINNT
System Directory       C:\WINNT\System32
Boot Device    \Device\Harddisk0\Partition1
Country Code   1
User Name      CLIENT1\Administrator
Time Zone      Pacific Daylight Time
Total Physical Memory  392752 kbytes
Available Physical Memory     301800 kbytes
Total Virtual Memory   2321108 kbytes
Available Virtual Memory      2055172 kbytes
Page File Space        1928356 kbytes
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  Port    Status
2       Standard floppy disk controller       0       OK

[Forced Hardware]

Device  PNP Device ID
No Forced Hardware

[I/O]

Address Range  Device  Status
0x0000-0x000F  Free    OK
0x0010-0x001F  PCI bus OK
0x0020-0x0021  Free    OK
0x0022-0x003F  PCI bus OK
0x0040-0x0043  Free    OK
0x0044-0x0047  PCI bus OK
0x0048-0x004B  Free    OK
0x004C-0x006F  PCI bus OK
0x0060-0x0060  PC/AT Enhanced PS/2 Keyboard (101/102-Key)   OK
0x0061-0x0061  System speaker OK
0x0064-0x0064  PC/AT Enhanced PS/2 Keyboard (101/102-Key)   OK
0x0070-0x0071  Free    OK
0x0072-0x007F  PCI bus OK
0x0080-0x008F  Free    OK
0x0090-0x0091  PCI bus OK
0x0092-0x0092  Free    OK
0x0093-0x009F  PCI bus OK
0x00A0-0x00A1  Free    OK
0x00A2-0x00BF  PCI bus OK
0x00C0-0x00CF  Free    OK
0x00D0-0x00EF  PCI bus OK
0x00F0-0x00FF  Free    OK
0x0100-0x0CF7  PCI bus OK
0x01F0-0x01F7  Primary IDE Channel    OK
0x0274-0x0277  ISAPNP Read Data Port  OK
0x0279-0x0279  ISAPNP Read Data Port  OK
0x02F8-0x02FF  Communications Port (COM2)    OK
0x0378-0x037F  Printer Port (LPT1)    OK
0x03B0-0x03BB  Cirrus Logic 5446 Compatible Graphics Adapter       OK
0x03C0-0x03DF  Cirrus Logic 5446 Compatible Graphics Adapter       OK
0x03F0-0x03F5  Standard floppy disk controller       OK
0x03F6-0x03F6  Primary IDE Channel    OK
0x03F7-0x03F7  Standard floppy disk controller       OK
0x03F8-0x03FF  Communications Port (COM1)    OK
0x0A79-0x0A79  ISAPNP Read Data Port  OK
0x0CF8-0x0CFF  Free    OK
```

```
0x0D00-0xFFFF  PCI bus OK
0xF800-0xF8FF  Adaptec AIC-7880 PCI SCSI Controller  OK
0xFC90-0xFC9F  Intel(r) 82371AB/EB PCI Bus Master IDE Controller    OK
0xFCA0-0xFCBF  HP NetServer 10/100TX PCI LAN Adapter #2     OK
0xFCC0-0xFCDF  HP NetServer 10/100TX PCI LAN Adapter OK
0xFCE0-0xFCFF  Intel 82371AB/EB PCI to USB Universal Host Controller
       OK

[IRQs]

IRQ Number     Device  Vector
1      PC/AT Enhanced PS/2 Keyboard (101/102-Key)   1
3      Communications Port (COM2)    3
4      Communications Port (COM1)    4
6      Standard floppy disk controller       6
12     Logitech PS/2 Port Mouse      12
14     Primary IDE Channel    14
19     Intel 82371AB/EB PCI to USB Universal Host Controller       19
32     HP NetServer 10/100TX PCI LAN Adapter 32
36     HP NetServer 10/100TX PCI LAN Adapter #2     36
40     Adaptec AIC-7880 PCI SCSI Controller  40

[Memory]

Range   Device  Memory Type    Status
0x000A0000-0x000BFFFF  PCI bus ReadWrite      OK
0x000A0000-0x000BFFFF  Cirrus Logic 5446 Compatible Graphics Adapter
       ReadWrite      OK
0x000C0000-0x000CCFFF  Unknown ReadOnly       OK
0x000CD000-0x000DFFFF  PCI bus ReadWrite      OK
0x000E0000-0x000FFFFF  Unknown ReadOnly       OK
0x18000000-0xFBFFFFFF  PCI bus ReadWrite      OK
0xFC000000-0xFDFFFFFF  PCI bus Prefetchable   OK
0xFC000000-0xFDFFFFFF  Cirrus Logic 5446 Compatible Graphics Adapter
       Prefetchable   OK
0xFEC00000-0xFEC003FF  Unknown ReadWrite      OK
0xFEC10000-0xFECFDFFF  PCI bus Prefetchable   OK
0xFECFC000-0xFECFCFFF  HP NetServer 10/100TX PCI LAN Adapter #2
       Prefetchable   OK
0xFECFD000-0xFECFDFFF  HP NetServer 10/100TX PCI LAN Adapter Prefetchable
       OK
0xFECFE000-0xFECFEFFF  PCI bus ReadWrite      OK
0xFECFE000-0xFECFEFFF  Cirrus Logic 5446 Compatible Graphics Adapter
       ReadWrite      OK
0xFECFF000-0xFECFFFFF  Adaptec AIC-7880 PCI SCSI Controller  ReadWrite
       OK
0xFED00000-0xFEDFFFFF  HP NetServer 10/100TX PCI LAN Adapter ReadWrite
       OK
0xFEE00000-0xFEE003FF  Unknown ReadWrite      OK
0xFEE01000-0xFFFEFFFF  PCI bus ReadWrite      OK
0xFFE00000-0xFFEFFFFF  HP NetServer 10/100TX PCI LAN Adapter #2
       ReadWrite      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 199680 bytes
       6/27/99 5:00:00 PM
c:\winnt\system32\lhacm.acm   Microsoft Corporation  Lernout And Hauspie
Codecs OK      C:\WINNT\System32\lhacm.acm   4.4.3385       34064 bytes
       8/23/99 6:33:26 AM
c:\winnt\system32\msadp32.acm Microsoft Corporation         OK
       C:\WINNT\System32\msadp32.acm 4.00    15120 bytes    6/27/99
5:00:00 PM
c:\winnt\system32\msg711.acm  Microsoft Corporation         OK
       C:\WINNT\System32\msg711.acm  4.00    10512 bytes    6/27/99
5:00:00 PM
c:\winnt\system32\imaadp32.acm        Microsoft Corporation         OK
       C:\WINNT\System32\imaadp32.acm        4.00    16144 bytes
       6/27/99 5:00:00 PM
c:\winnt\system32\tssoft32.acm        DSP GROUP, INC.               OK
       C:\WINNT\System32\tssoft32.acm        1.01    9488 bytes
       6/27/99 5:00:00 PM
c:\winnt\system32\msgsm32.acm Microsoft Corporation         OK
       C:\WINNT\System32\msgsm32.acm 4.00    22800 bytes    6/27/99
5:00:00 PM
c:\winnt\system32\msg723.acm  Microsoft Corporation         OK
       C:\WINNT\System32\msg723.acm  4.4.3385       109328 bytes
       8/23/99 6:33:25 AM

[Video Codecs]

Codec   Manufacturer   Description    Status  File    Version Size
       Creation Date
c:\winnt\system32\msh263.drv  Microsoft Corporation         OK
       C:\WINNT\System32\msh263.drv  4.4.3385       258320 bytes
       8/23/99 6:32:53 AM
c:\winnt\system32\msh261.drv  Microsoft Corporation         OK
       C:\WINNT\System32\msh261.drv  4.4.3385       167696 bytes
       8/23/99 6:33:25 AM
c:\winnt\system32\msvidc32.dll        Microsoft Corporation         OK
       C:\WINNT\System32\msvidc32.dll        5.00.2051.1    27920 bytes
       6/27/99 5:00:00 PM
c:\winnt\system32\iccvid.dll  Radius Inc.            OK
       C:\WINNT\System32\iccvid.dll  1.10.0.6       110592 bytes
       6/27/99 5:00:00 PM
c:\winnt\system32\msrle32.dll Microsoft Corporation         OK
       C:\WINNT\System32\msrle32.dll 5.00.2051.1    11024 bytes
       6/27/99 5:00:00 PM
c:\winnt\system32\ir50_32.dll Intel Corporation      Indeo® video 5.10
       OK      C:\WINNT\System32\ir50_32.dll R.5.10.15.2.54 755200 bytes
       6/27/99 5:00:00 PM
c:\winnt\system32\ir32_32.dll Intel(R) Corporation          OK
       C:\WINNT\System32\ir32_32.dll Not Available  199168 bytes
       6/27/99 5:00:00 PM

[CD-ROM]
```

```
Item    Value
Drive   D:
Description     CD-ROM Drive
Media Loaded    False
Media Type      CD-ROM
Name    TEAC CD-532E-B
Manufacturer    (Standard CD-ROM drives)
Status  Unknown
Transfer Rate   Not Available
SCSI Target ID 0
PNP Device ID   IDE\CDROMTEAC_CD-532E-
B_____________________________2.1A____\4&1C9CBC13&0&0.0.0


[Sound Device]

Item    Value
No sound devices

[Display]

Item    Value
Name    Current Video Configuration
Adapter Type    Cirrus Logic 5446BE, CIRRUS compatible
Adapter Description     Cirrus Logic Compatible
Adapter RAM     1048576 bytes
Installed Drivers       vga,cirrus,vga256,vga64k
Color Resolution        18 bits/pixel
Color Planes    1
Color Table Entries     20
Resolution      800 x 600 x 60 hertz
Bits/Pixel      8
Pixels/Logical X Inch   96
Pixels/Logical Y Inch   96
Screen Height   240 pixels
Screen Width    320 pixels


[Infrared]

Item    Value
No infrared devices

[Input]

[ Following are sub-categories of this main category ]

[Keyboard]

Item    Value
Description     PC/AT Enhanced PS/2 Keyboard (101/102-Key)
Name    Enhanced (101- or 102-key)
Layout  00000409
PNP Device ID   ROOT\*PNP030B\1_0_22_0_32_0
NumberOfFunctionKeys    12


[Pointing Device]

Item    Value
Hardware Type   Logitech PS/2 Port Mouse
Number of Buttons       3
Status  OK
PNP Device ID   ROOT\*PNP0F12\1_0_21_0_31_0
Power Management Supported      False
Double Click Threshold 6
Handedness      2


[Modem]

Item    Value
No modems

[Network]

[ Following are sub-categories of this main category ]

[Adapter]

Item    Value
Name    [000] HP NetServer 10/100TX PCI LAN Adapter
Adapter Type    pci\ven_8086&dev_1229&subsys_10c3103c
Product Name    HP NetServer 10/100TX PCI LAN Adapter
Installed       True
PNP Device ID   PCI\VEN_8086&DEV_1229&SUBSYS_10C3103C&REV_05\2&EBB567F&0&40
Last Reset      10/21/99 2:15:59 AM
Index   0
Service Name    HPTX
IP Address      192.168.91.1
IP Subnet       255.255.255.0
Default IP Gateway
DHCP Enabled    False
DHCP Server     Not Available
DHCP Lease Expires      Not Available
DHCP Lease Obtained     Not Available
MAC Address     00:90:27:8A:EA:88
Service Name    HPTX
IRQ Number      32
I/O Port        0xFCC0-0xFCDF
Driver  c:\winnt\system32\drivers\hptxnt5.sys (56080 bytes, 3.37.15.0016)

Name    [001] HP NetServer 10/100TX PCI LAN Adapter
Adapter Type    pci\ven_8086&dev_1229&subsys_10c3103c
Product Name    HP NetServer 10/100TX PCI LAN Adapter
Installed       True
PNP Device ID   PCI\VEN_8086&DEV_1229&SUBSYS_10C3103C&REV_05\2&EBB567F&0&48
Last Reset      10/21/99 2:15:59 AM
Index   1
Service Name    HPTX
IP Address      192.168.10.1
IP Subnet       255.255.255.0
Default IP Gateway
DHCP Enabled    False
DHCP Server     Not Available
DHCP Lease Expires      Not Available
DHCP Lease Obtained     Not Available
MAC Address     00:90:27:8A:EB:D5
```

```
Service Name   HPTX
IRQ Number     36
I/O Port       0xFCA0-0xFCBF
Driver  c:\winnt\system32\drivers\hptxnt5.sys (56080 bytes, 3.37.15.0016)

Name    [002] RAS Async Adapter
Adapter Type   sw\{eeab7790-c514-11d1-b42b-00805fc1270e}
Product Name   RAS Async Adapter
Installed      True
PNP Device ID  Not Available
Last Reset     10/21/99 2:15:59 AM
Index  2
Service Name   AsyncMac
IP Address     Not Available
IP Subnet      Not Available
Default IP Gateway     Not Available
DHCP Enabled   False
DHCP Server    Not Available
DHCP Lease Expires     Not Available
DHCP Lease Obtained    Not Available
MAC Address    Not Available
Service Name   Not Available

Name    [003] WAN Miniport (L2TP)
Adapter Type   ms_l2tpminiport
Product Name   WAN Miniport (L2TP)
Installed      True
PNP Device ID  ROOT\MS_L2TPMINIPORT\0000
Last Reset     10/21/99 2:15:59 AM
Index  3
Service Name   Rasl2tp
IP Address     Not Available
IP Subnet      Not Available
Default IP Gateway     Not Available
DHCP Enabled   False
DHCP Server    Not Available
DHCP Lease Expires     Not Available
DHCP Lease Obtained    Not Available
MAC Address    Not Available
Service Name   Rasl2tp
Driver  c:\winnt\system32\drivers\rasl2tp.sys (47216 bytes, 5.00.2059.1)

Name    [004] WAN Miniport (PPTP)
Adapter Type   ms_pptpminiport
Product Name   WAN Miniport (PPTP)
Installed      True
PNP Device ID  ROOT\MS_PPTPMINIPORT\0000
Last Reset     10/21/99 2:15:59 AM
Index  4
Service Name   PptpMiniport
IP Address     Not Available
IP Subnet      Not Available
Default IP Gateway     Not Available
DHCP Enabled   False
DHCP Server    Not Available
DHCP Lease Expires     Not Available
DHCP Lease Obtained    Not Available
MAC Address    Not Available
Service Name   PptpMiniport
Driver  c:\winnt\system32\drivers\raspptp.sys (45360 bytes, 5.00.2069.1)

Name    [005] Direct Parallel
Adapter Type   ms_ptiminiport
Product Name   Direct Parallel
Installed      True
PNP Device ID  ROOT\MS_PTIMINIPORT\0000
Last Reset     10/21/99 2:15:59 AM
Index  5
Service Name   Raspti
IP Address     Not Available
IP Subnet      Not Available
Default IP Gateway     Not Available
DHCP Enabled   False
DHCP Server    Not Available
DHCP Lease Expires     Not Available
DHCP Lease Obtained    Not Available
MAC Address    Not Available
Service Name   Raspti
Driver  c:\winnt\system32\drivers\raspti.sys (16656 bytes, 5.00.2062.1)

Name    [006] WAN Miniport (IP)
Adapter Type   ms_ndiswanip
Product Name   WAN Miniport (IP)
Installed      True
PNP Device ID  ROOT\MS_NDISWANIP\0000
Last Reset     10/21/99 2:15:59 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 (92176 bytes, 5.00.2065.1)

Name    [007] WAN Miniport (NetBEUI, Dial In)
Adapter Type   ms_ndiswannbfin
Product Name   WAN Miniport (NetBEUI, Dial In)
Installed      True
PNP Device ID  ROOT\MS_NDISWANNBFIN\0000
Last Reset     10/21/99 2:15:59 AM
Index  7
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 (92176 bytes, 5.00.2065.1)
```

```
Name    [008] WAN Miniport (NetBEUI, Dial In)
Adapter Type    ms_ndiswannbfin
Product Name    WAN Miniport (NetBEUI, Dial In)
Installed       True
PNP Device ID   ROOT\MS_NDISWANNBFIN\0001
Last Reset      10/21/99 2:15:59 AM
Index   8
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 (92176 bytes, 5.00.2065.1)

Name    [009] WAN Miniport (NetBEUI, Dial Out)
Adapter Type    ms_ndiswannbfout
Product Name    WAN Miniport (NetBEUI, Dial Out)
Installed       True
PNP Device ID   ROOT\MS_NDISWANNBFOUT\0000
Last Reset      10/21/99 2:15:59 AM
Index   9
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 (92176 bytes, 5.00.2065.1)


[Protocol]

Item    Value
Name    MSAFD Tcpip [TCP/IP]
ConnectionlessService  False
GuaranteesDelivery     True
GuaranteesSequencing   True
MaximumAddressSize     16 bytes
MaximumMessageSize     0 bytes
MessageOriented        False
MinimumAddressSize     16 bytes
PseudoStreamOriented   False
SupportsBroadcasting   False
SupportsConnectData    False
SupportsDisconnectData False
SupportsEncryption     False
SupportsExpeditedData  True
SupportsFragmentation  False
SupportsGracefulClosing        True
SupportsGuaranteedBandwidth    False
SupportsMulticasting   False

Name    MSAFD Tcpip [UDP/IP]
ConnectionlessService  True
GuaranteesDelivery     False
GuaranteesSequencing   False
MaximumAddressSize     16 bytes
MaximumMessageSize     65467 bytes
MessageOriented        True
MinimumAddressSize     16 bytes
PseudoStreamOriented   False
SupportsBroadcasting   True
SupportsConnectData    False
SupportsDisconnectData False
SupportsEncryption     False
SupportsExpeditedData  False
SupportsFragmentation  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
SupportsFragmentation  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
SupportsFragmentation  False
SupportsGracefulClosing        True
SupportsGuaranteedBandwidth    False
SupportsMulticasting   False
```

```
Name    MSAFD NetBIOS [\Device\Nbf_{866B1FFE-5C50-488C-9265-63F01B40EE90}]
SEQPACKET 0
ConnectionlessService  False
GuaranteesDelivery     True
GuaranteesSequencing   True
MaximumAddressSize     20 bytes
MaximumMessageSize     64000 bytes
MessageOriented        True
MinimumAddressSize     20 bytes
PseudoStreamOriented   False
SupportsBroadcasting   False
SupportsConnectData    False
SupportsDisconnectData False
SupportsEncryption     False
SupportsExpeditedData  False
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting   False

Name    MSAFD NetBIOS [\Device\Nbf_{866B1FFE-5C50-488C-9265-63F01B40EE90}]
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
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting   False

Name    MSAFD NetBIOS [\Device\Nbf_{ECC929FD-0C89-465C-AF2E-98C31F5F54F9}]
SEQPACKET 1
ConnectionlessService  False
GuaranteesDelivery     True
GuaranteesSequencing   True
MaximumAddressSize     20 bytes
MaximumMessageSize     64000 bytes
MessageOriented        True
MinimumAddressSize     20 bytes
PseudoStreamOriented   False
SupportsBroadcasting   False
SupportsConnectData    False
SupportsDisconnectData False
SupportsEncryption     False
SupportsExpeditedData  False
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting   False

Name    MSAFD NetBIOS [\Device\Nbf_{ECC929FD-0C89-465C-AF2E-98C31F5F54F9}]
DATAGRAM 1
ConnectionlessService  True
GuaranteesDelivery     False
GuaranteesSequencing   False
MaximumAddressSize     20 bytes
MaximumMessageSize     64000 bytes
MessageOriented        True
MinimumAddressSize     20 bytes
PseudoStreamOriented   False
SupportsBroadcasting   True
SupportsConnectData    False
SupportsDisconnectData False
SupportsEncryption     False
SupportsExpeditedData  False
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting   False

Name    MSAFD NetBIOS [\Device\Nbf_NdisWanNbfIn{04FDB3B5-F73C-47B3-BC97-
364F543F7D5A}] SEQPACKET 2
ConnectionlessService  False
GuaranteesDelivery     True
GuaranteesSequencing   True
MaximumAddressSize     20 bytes
MaximumMessageSize     64000 bytes
MessageOriented        True
MinimumAddressSize     20 bytes
PseudoStreamOriented   False
SupportsBroadcasting   False
SupportsConnectData    False
SupportsDisconnectData False
SupportsEncryption     False
SupportsExpeditedData  False
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting   False

Name    MSAFD NetBIOS [\Device\Nbf_NdisWanNbfIn{04FDB3B5-F73C-47B3-BC97-
364F543F7D5A}] DATAGRAM 2
ConnectionlessService  True
GuaranteesDelivery     False
GuaranteesSequencing   False
MaximumAddressSize     20 bytes
MaximumMessageSize     64000 bytes
MessageOriented        True
MinimumAddressSize     20 bytes
PseudoStreamOriented   False
SupportsBroadcasting   True
SupportsConnectData    False
SupportsDisconnectData False
SupportsEncryption     False
SupportsExpeditedData  False
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
```

```
SupportsMulticasting   False

Name    MSAFD NetBIOS [\Device\Nbf_NdisWanNbfIn{735028B2-D6B4-4FA2-A138-
432BC85E4FF0}] SEQPACKET 3
ConnectionlessService  False
GuaranteesDelivery     True
GuaranteesSequencing   True
MaximumAddressSize     20 bytes
MaximumMessageSize     64000 bytes
MessageOriented        True
MinimumAddressSize     20 bytes
PseudoStreamOriented   False
SupportsBroadcasting   False
SupportsConnectData    False
SupportsDisconnectData False
SupportsEncryption     False
SupportsExpeditedData  False
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting   False

Name    MSAFD NetBIOS [\Device\Nbf_NdisWanNbfIn{735028B2-D6B4-4FA2-A138-
432BC85E4FF0}] 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
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting   False

Name    MSAFD NetBIOS [\Device\Nbf_NdisWanNbfOut{9A1C2258-4CA9-46A4-9550-
69F23E8F3539}] 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
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting   False

Name    MSAFD NetBIOS [\Device\Nbf_NdisWanNbfOut{9A1C2258-4CA9-46A4-9550-
69F23E8F3539}] 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
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting   False

Name    MSAFD NetBIOS [\Device\NetBT_Tcpip_{866B1FFE-5C50-488C-9265-
63F01B40EE90}] 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
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting   False

Name    MSAFD NetBIOS [\Device\NetBT_Tcpip_{866B1FFE-5C50-488C-9265-
63F01B40EE90}] 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
SupportsFragmentation  False
```

```
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting  False

Name    MSAFD NetBIOS [\Device\NetBT_Tcpip_{ECC929FD-0C89-465C-AF2E-
98C31F5F54F9}] 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
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting  False

Name    MSAFD NetBIOS [\Device\NetBT_Tcpip_{ECC929FD-0C89-465C-AF2E-
98C31F5F54F9}] 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
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting  False

Name    MSAFD NetBIOS [\Device\NetBT_Tcpip_{EE8FF43E-80E6-41B0-9040-
81062CA1A36E}] SEQPACKET 7
ConnectionlessService  False
GuaranteesDelivery     True
GuaranteesSequencing   True
MaximumAddressSize     20 bytes
MaximumMessageSize     64000 bytes
MessageOriented        True
MinimumAddressSize     20 bytes
PseudoStreamOriented   False
SupportsBroadcasting   False
SupportsConnectData    False
SupportsDisconnectData False
SupportsEncryption     False
SupportsExpeditedData  False
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting  False

Name    MSAFD NetBIOS [\Device\NetBT_Tcpip_{EE8FF43E-80E6-41B0-9040-
81062CA1A36E}] DATAGRAM 7
ConnectionlessService  True
GuaranteesDelivery     False
GuaranteesSequencing   False
MaximumAddressSize     20 bytes
MaximumMessageSize     64000 bytes
MessageOriented        True
MinimumAddressSize     20 bytes
PseudoStreamOriented   False
SupportsBroadcasting   True
SupportsConnectData    False
SupportsDisconnectData False
SupportsEncryption     False
SupportsExpeditedData  False
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting  False

Name    MSAFD NetBIOS [\Device\NetBT_Tcpip_{99867619-09B4-4E52-A092-
F8E370FB9A6C}] SEQPACKET 8
ConnectionlessService  False
GuaranteesDelivery     True
GuaranteesSequencing   True
MaximumAddressSize     20 bytes
MaximumMessageSize     64000 bytes
MessageOriented        True
MinimumAddressSize     20 bytes
PseudoStreamOriented   False
SupportsBroadcasting   False
SupportsConnectData    False
SupportsDisconnectData False
SupportsEncryption     False
SupportsExpeditedData  False
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting  False

Name    MSAFD NetBIOS [\Device\NetBT_Tcpip_{99867619-09B4-4E52-A092-
F8E370FB9A6C}] DATAGRAM 8
ConnectionlessService  True
GuaranteesDelivery     False
GuaranteesSequencing   False
MaximumAddressSize     20 bytes
MaximumMessageSize     64000 bytes
MessageOriented        True
MinimumAddressSize     20 bytes
PseudoStreamOriented   False
SupportsBroadcasting   True
SupportsConnectData    False
SupportsDisconnectData False
SupportsEncryption     False
```

```
SupportsExpeditedData  False
SupportsFragmentation  False
SupportsGracefulClosing       False
SupportsGuaranteedBandwidth   False
SupportsMulticasting   False


[WinSock]

Item    Value
File    c:\winnt\system32\winsock.dll
Version 3.10
Size    2864 bytes

File    c:\winnt\system32\wsock32.dll
Version 5.00.2066.1
Size    21776 bytes


[Ports]

[ Following are sub-categories of this main category ]

[Serial]

Item    Value
Name    Communications Port (COM2)
Status  Error
PNP Device ID  ROOT\*PNP0501\1_0_17_1_0_0
Maximum Input Buffer Size     0
Maximum Output Buffer Size    False
Settable Baud Rate     True
Settable Data Bits     True
Settable Flow Control  True
Settable Parity        True
Settable Parity Check  True
Settable Stop Bits     True
Settable RLSD  True
Supports RLSD  True
Supports 16 Bit Mode   False
Supports Special Characters   False
Baud Rate      9600
Bits/Byte      8
Stop Bits      1
Parity  NONE
Busy    0
Abort Read/Write on Error     0
Binary Mode Enabled    -1
Continue XMit on XOff  0
CTS Outflow Control    0
Discard NULL Bytes     0
DSR Outflow Control    0
DSR Sensitivity        0
DTR Flow Control Type  ENABLE
EOF Character  0
Error Replace Character       0
Error Replacement Enabled     0
Event Character        0
Parity Check Enabled   0
RTS Flow Control Type  ENABLE
XOff Character 19
XOffXMit Threshold     512
XOn Character  17
XOnXMit Threshold      2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control       0
IRQ Number     3
I/O Port       0x02F8-0x02FF
Driver  c:\winnt\system32\drivers\serial.sys (62384 bytes, 5.00.2051.1)

Name    Communications Port (COM1)
Status  Error
PNP Device ID  ROOT\*PNP0501\PNPBIOS_14
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 (62384 bytes, 5.00.2051.1)

Name    Communications Port (COM3)
Status  OK
PNP Device ID  ROOT\*PNP0501\PNPBIOS_22
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      Not Available
Bits/Byte      Not Available
Stop Bits      Not Available
Parity  Not Available
Busy    0
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   Not Available
RTS Flow Control Type  Not Available
XOff Character Not Available
XOffXMit Threshold     Not Available
XOn Character  Not Available
XOnXMit Threshold      Not Available
XOnXOff InFlow Control Not Available
XOnXOff OutFlow Control       Not Available
Driver  c:\winnt\system32\drivers\serial.sys (62384 bytes, 5.00.2051.1)


[Parallel]

Item    Value
Name    LPT1
PNP Device ID  ROOT\*PNP0400\PNPBIOS_13


[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    Not Available
Free Space     1884553216 bytes
Volume Name
Volume Serial Number   E4A951E7
Partition      Disk #0, Partition #0
Partition Size 4252437504 bytes
Starting Offset        32256 bytes
Drive Description      SCSI Fixed Disk
Drive Manufacturer     HP
Drive Model    Not Available
Drive BytesPerSector   512
Drive MediaLoaded      True
Drive MediaType        Fixed hard disk media
Drive Partitions       1
Drive SCSIBus  0
Drive SCSILogicalUnit  0
Drive SCSIPort 1
Drive SCSITargetId     0
Drive SectorsPerTrack  63
Drive Size     4260695040 bytes
Drive TotalCylinders   518
Drive TotalSectors     8321670
Drive TotalTracks      132090
Drive TracksPerCylinder       255


[SCSI]

Item    Value
Name    Adaptec AIC-7880 PCI SCSI Controller
Caption Adaptec AIC-7880 PCI SCSI Controller
Driver  Not Available
Status  OK
PNP Device ID  PCI\VEN_9004&DEV_8078&SUBSYS_78809004&REV_01\2&EBB567F&0&50
Device ID      PCI\VEN_9004&DEV_8078&SUBSYS_78809004&REV_01\2&EBB567F&0&50
Device Map     Not Available
Index   Not Available
Max Number Controlled  Not Available
IRQ Number     40
I/O Port       0xF800-0xF8FF
Driver  c:\winnt\system32\drivers\aic78xx.sys (56784 bytes, v2.20b)


[Printing]

Name    Port Name      Server Name
No printing information

[Problem Devices]

Device  PNP Device ID  Error Code
Communications Port (COM3)    ROOT\*PNP0501\PNPBIOS_22      29

[USB]

Device  PNP Device ID
USB Root Hub   USB\ROOT_HUB\3&29761208&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
acpi    ACPI    Not Available  Kernel Driver  False   Disabled
        Stopped OK      Normal  False   False
acpiec  ACPIEC  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
asc3550 asc3550 Not Available  Kernel Driver  False   Disabled
        Stopped OK      Normal  False   False
asyncmac        RAS Asynchronous Media Driver
        c:\winnt\system32\drivers\asyncmac.sys       Kernel Driver  False
        Manual  Stopped OK      Normal  False   False
atapi   Standard IDE/ESDI Hard Disk Controller
        c:\winnt\system32\drivers\atapi.sys   Kernel Driver  True    Boot
        Running OK      Normal  False   True
atdisk  Atdisk  Not Available  Kernel Driver  False   Disabled
        Stopped OK      Ignore  False   False
atmarpc ATM ARP Client Protocol
        c:\winnt\system32\drivers\atmarpc.sys Kernel Driver  False   Manual
        Stopped OK      Normal  False   False
audstub Audio Stub Driver     c:\winnt\system32\drivers\audstub.sys Kernel
Driver  True    Manual  Running OK      Normal  False   True
beep    System speaker driver  c:\winnt\system32\drivers\beep.sys    Kernel
Driver  True    Manual  Running OK      Normal  False   True
buslogic        BusLogic        Not Available  Kernel Driver  False
        Disabled        Stopped OK      Normal  False   False
cdaudio Cdaudio c:\winnt\system32\drivers\cdaudio.sys Kernel Driver  False
        System  Stopped OK      Ignore  False   False
cdfs    Cdfs    c:\winnt\system32\drivers\cdfs.sys    File System Driver
        True    Disabled        Running OK      Normal  False   True
cdrom   CD-ROM Driver  c:\winnt\system32\drivers\cdrom.sys   Kernel Driver
        True    System  Running OK      Normal  False   True
changer Changer Not Available  Kernel Driver  False   System  Stopped OK
        Ignore  False   False
cirrus  cirrus  c:\winnt\system32\drivers\cirrus.sys  Kernel Driver  True
        Manual  Running OK      Ignore  False   True
cpqarray        Cpqarray        Not Available  Kernel Driver  False
        Disabled        Stopped OK      Normal  False   False
cpqfcalm        cpqfcalm        Not Available  Kernel Driver  False
        Disabled        Stopped OK      Normal  False   False
cpqfws2e        cpqfws2e        Not Available  Kernel Driver  False
        Disabled        Stopped OK      Normal  False   False
dac960nt        dac960nt        Not Available  Kernel Driver  False
        Disabled        Stopped OK      Normal  False   False
deckzpsx        deckzpsx        Not Available  Kernel Driver  False
        Disabled        Stopped OK      Normal  False   False
dfsdriver       DfsDriver       c:\winnt\system32\drivers\dfs.sys     File
System Driver  True    Boot    Running OK      Normal  False   True
disk    Disk Driver    c:\winnt\system32\drivers\disk.sys    Kernel Driver
        True    Boot    Running OK      Normal  False   True
diskperf        Diskperf        c:\winnt\system32\drivers\diskperf.sys
        Kernel Driver  True    Boot    Running OK      Normal  False   True
dmboot  dmboot  c:\winnt\system32\drivers\dmboot.sys  Kernel Driver  False
        Disabled        Stopped OK      Normal  False   False
dmio    Logical Disk Manager Driver   c:\winnt\system32\drivers\dmio.sys
        Kernel Driver  True    Boot    Running OK      Normal  False   True
dmload  dmload  c:\winnt\system32\drivers\dmload.sys  Kernel Driver  True
        Boot    Running OK      Normal  False   True
efs     EFS     c:\winnt\system32\drivers\efs.sys     File System Driver
        True    Disabled        Running OK      Normal  False   True
fastfat Fastfat c:\winnt\system32\drivers\fastfat.sys File System Driver
        True    Disabled        Running OK      Normal  False   True
fd16_700        Fd16_700        Not Available  Kernel Driver  False
        Disabled        Stopped OK      Normal  False   False
fdc     Floppy Disk Controller Driver c:\winnt\system32\drivers\fdc.sys
        Kernel Driver  True    Manual  Running OK      Normal  False   True
fireport        fireport        Not Available  Kernel Driver  False
        Disabled        Stopped OK      Normal  False   False
flashpnt        flashpnt        Not Available  Kernel Driver  False
        Disabled        Stopped OK      Normal  False   False
flpydisk        Floppy Disk Driver
        c:\winnt\system32\drivers\flpydisk.sys       Kernel Driver  True
        Manual  Running OK      Normal  False   True
ftdisk  Volume Manager Driver  c:\winnt\system32\drivers\ftdisk.sys  Kernel
Driver  True    Boot    Running OK      Normal  False   True
gpc     Generic Packet Classifier     c:\winnt\system32\drivers\msgpc.sys
        Kernel Driver  True    Manual  Running OK      Normal  False   True
hptx    HP 10/100TX PCI LAN Adapter NT Driver
        c:\winnt\system32\drivers\hptxnt5.sys Kernel Driver  True    Manual
        Running OK      Normal  False   True
i8042prt        i8042 Keyboard and PS/2 Mouse Port Driver
        c:\winnt\system32\drivers\i8042prt.sys       Kernel Driver  True
        System  Running OK      Normal  False   True
ini910u ini910u Not Available  Kernel Driver  False   Disabled
        Stopped OK      Normal  False   False
intelide        IntelIde        c:\winnt\system32\drivers\intelide.sys
        Kernel Driver  True    Boot    Running OK      Normal  False   True
ipfilterdriver IP Traffic Filter Driver
        c:\winnt\system32\drivers\ipfltdrv.sys       Kernel Driver  False
        Manual  Stopped OK      Normal  False   False
ipinip  IP in IP Tunnel Driver c:\winnt\system32\drivers\ipinip.sys  Kernel
Driver  False   Manual  Stopped OK      Normal  False   False
```

```
ipnat   IP Network Address Translator c:\winnt\system32\drivers\ipnat.sys
        Kernel Driver  False   Manual  Stopped OK      Normal  False   False
ipsec   IPSEC driver   c:\winnt\system32\drivers\ipsec.sys   Kernel Driver
        False   Manual  Stopped OK      Normal  False   False
ipsraidn        ipsraidn        Not Available  Kernel Driver  False
        Disabled        Stopped OK      Normal  False   False
isapnp  PnP ISA/EISA Bus Driver         c:\winnt\system32\drivers\isapnp.sys
        Kernel Driver  True    Boot    Running OK      Critical        False
        True
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
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
mup     Mup     c:\winnt\system32\drivers\mup.sys     File System Driver
        True    Boot    Running OK      Normal  False   True
nbf     NetBEUI Protocol        c:\winnt\system32\drivers\nbf.sys     Kernel
Driver  True    Auto    Running OK      Normal  False   True
ncrc710 Ncrc710 Not Available  Kernel Driver  False   Disabled
        Stopped OK      Normal  False   False
ndis    NDIS System Driver     c:\winnt\system32\drivers\ndis.sys    Kernel
Driver  True    Boot    Running OK      Normal  False   True
ndistapi        Remote Access NDIS TAPI Driver
        c:\winnt\system32\drivers\ndistapi.sys        Kernel Driver  True
        Manual  Running OK      Normal  False   True
ndiswan Remote Access NDIS WAN Driver
        c:\winnt\system32\drivers\ndiswan.sys Kernel Driver  True    Manual
        Running OK      Normal  False   True
ndproxy NDIS Proxy      c:\winnt\system32\drivers\ndproxy.sys Kernel Driver
        True    Manual  Running OK      Normal  False   True
netbios NetBIOS Interface      c:\winnt\system32\drivers\netbios.sys File
System Driver  True    System  Running OK      Normal  False   True
netbt   NetBios over Tcpip     c:\winnt\system32\drivers\netbt.sys   Kernel
Driver  True    System  Running OK      Normal  False   True
netdetect       NetDetect       c:\winnt\system32\drivers\netdtect.sys
        Kernel Driver  False   Manual  Stopped OK      Normal  False   False
npfs    Npfs    c:\winnt\system32\drivers\npfs.sys    File System Driver
        True    System  Running OK      Normal  False   True
ntfs    Ntfs    c:\winnt\system32\drivers\ntfs.sys    File System Driver
        True    Disabled        Running OK      Normal  False   True
null    Null    c:\winnt\system32\drivers\null.sys    Kernel Driver  True
        System  Running OK      Normal  False   True
nwlnkflt        IPX Traffic Filter Driver
        c:\winnt\system32\drivers\nwlnkflt.sys        Kernel Driver  False
        Manual  Stopped OK      Normal  False   False
nwlnkfwd        IPX Traffic Forwarder Driver
        c:\winnt\system32\drivers\nwlnkfwd.sys        Kernel Driver  False
        Manual  Stopped OK      Normal  False   False
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  Not Available  Kernel Driver  False   Disabled
        Stopped OK      Normal  False   False
pcmcia  Pcmcia  c:\winnt\system32\drivers\pcmcia.sys  Kernel Driver  False
        Disabled        Stopped OK      Normal  False   False
pdcomp  PDCOMP  Not Available  Kernel Driver  False   Manual  Stopped OK
        Ignore  False   False
pdframe PDFRAME Not Available  Kernel Driver  False   Manual  Stopped OK
        Ignore  False   False
pdreli  PDRELI  Not Available  Kernel Driver  False   Manual  Stopped OK
        Ignore  False   False
pdrframe        PDRFRAME        Not Available  Kernel Driver  False   Manual
        Stopped OK      Ignore  False   False
pptpminiport    WAN Miniport (PPTP)
        c:\winnt\system32\drivers\raspptp.sys Kernel Driver  True    Manual
        Running OK      Normal  False   True
ptilink Direct Parallel Link Driver
        c:\winnt\system32\drivers\ptilink.sys Kernel Driver  True    Manual
        Running OK      Normal  False   True
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    Auto
        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
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
scsiscan        Scsiscan        Not Available  Kernel Driver  False   System
        Stopped OK      Ignore  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
simbad  Simbad  Not Available  Kernel Driver  False   Disabled
        Stopped OK      Normal  False   False
sparrow Sparrow Not Available  Kernel Driver  False   Disabled
        Stopped OK      Normal  False   False
spud    Special Purpose Utility Driver
        c:\winnt\system32\drivers\spud.sys    Kernel Driver  True    Manual
        Running OK      Normal  False   True
srv     Srv     c:\winnt\system32\drivers\srv.sys     File System Driver
        True    Manual  Running OK      Normal  False   True
swenum  Software Bus Driver    c:\winnt\system32\drivers\swenum.sys  Kernel
Driver  True    Manual  Running OK      Normal  False   True
symc810 symc810 Not Available  Kernel Driver  False   Disabled
        Stopped OK      Normal  False   False
symc8xx symc8xx Not Available  Kernel Driver  False   Disabled
        Stopped OK      Normal  False   False
sym_hi  sym_hi  Not Available  Kernel Driver  False   Disabled
        Stopped OK      Normal  False   False
tcpip   TCP/IP Protocol Driver c:\winnt\system32\drivers\tcpip.sys   Kernel
Driver  True    System  Running OK      Normal  False   True
tdasync TDASYNC c:\winnt\system32\drivers\tdasync.sys Kernel Driver  False
        Manual  Stopped OK      Ignore  False   False
tdipx   TDIPX   c:\winnt\system32\drivers\tdipx.sys   Kernel Driver  False
        Manual  Stopped OK      Ignore  False   False
tdnetb  TDNETB  c:\winnt\system32\drivers\tdnetb.sys  Kernel Driver  False
        Manual  Stopped OK      Ignore  False   False
tdpipe  TDPIPE  c:\winnt\system32\drivers\tdpipe.sys  Kernel Driver  False
        Manual  Stopped OK      Ignore  False   False
tdspx   TDSPX   c:\winnt\system32\drivers\tdspx.sys   Kernel Driver  False
        Manual  Stopped OK      Ignore  False   False
tdtcp   TDTCP   c:\winnt\system32\drivers\tdtcp.sys   Kernel Driver  False
        Manual  Stopped OK      Ignore  False   False
termdd  Terminal Device Driver c:\winnt\system32\drivers\termdd.sys  Kernel
Driver  False   Disabled        Stopped OK      Normal  False   False
tga     tga     Not Available  Kernel Driver  False   System  Stopped OK
        Ignore  False   False
udfs    Udfs    c:\winnt\system32\drivers\udfs.sys    File System Driver
        False   Disabled        Stopped OK      Normal  False   False
uhcd    Microsoft USB Universal Host Controller Driver
        c:\winnt\system32\drivers\uhcd.sys    Kernel Driver  True    Auto
        Running OK      Normal  False   True
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    Auto
        Running OK      Normal  False   True
vgasave VgaSave c:\winnt\system32\drivers\vga.sys     Kernel Driver  False
        System  Stopped OK      Ignore  False   False
wanarp  Remote Access IP ARP Driver    c:\winnt\system32\drivers\wanarp.sys
        Kernel Driver  True    Manual  Running OK      Normal  False   True
wdica   WDICA   Not Available  Kernel Driver  False   Manual  Stopped OK
        Ignore  False   False

[Environment Variables]

Variable        Value   User Name
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
Os2LibPath      %SystemRoot%\system32\os2\dll;        <SYSTEM>
Path
        %SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\M
SSQL7\BINN      <SYSTEM>
windir  %SystemRoot%    <SYSTEM>
OS      Windows_NT      <SYSTEM>
PROCESSOR_ARCHITECTURE x86      <SYSTEM>
PROCESSOR_LEVEL         6       <SYSTEM>
PROCESSOR_IDENTIFIER    x86 Family 6 Model 7 Stepping 3, GenuineIntel
        <SYSTEM>
PROCESSOR_REVISION      0703    <SYSTEM>
NUMBER_OF_PROCESSORS    1       <SYSTEM>
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WS;.WSH      <SYSTEM>
TEMP    %SystemRoot%\TEMP       <SYSTEM>
TMP     %SystemRoot%\TEMP       <SYSTEM>
TEMP    %USERPROFILE%\Local Settings\Temp     CLIENT1\Administrator
TMP     %USERPROFILE%\Local Settings\Temp     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

[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       204800  1413120 Not Available
        Unknown Unknown Unknown
smss.exe        c:\winnt\system32\smss.exe     148     11      204800
        1413120 10/21/99 9:16:15 AM     5.00.2054.1     42768 bytes
        6/27/99 5:00:00 PM
csrss.exe       Not Available   176     13      Not Available   Not Available
        10/21/99 9:16:22 AM     Unknown Unknown Unknown
winlogon.exe    c:\winnt\system32\winlogon.exe         196     13      204800
        1413120 10/21/99 9:16:23 AM     5.00.2067.1     158992 bytes
        6/27/99 5:00:00 PM
services.exe    c:\winnt\system32\services.exe         224     9       204800
        1413120 10/21/99 9:16:25 AM     5.00.2063.1     87824 bytes
        6/27/99 5:00:00 PM
```

```
lsass.exe      c:\winnt\system32\lsass.exe   236     13      204800
       1413120 10/21/99 9:16:25 AM    5.00.2065.1    33552 bytes
       6/27/99 5:00:00 PM
svchost.exe    c:\winnt\system32\svchost.exe 368     8       204800
       1413120 10/21/99 9:16:29 AM    5.00.2051.1    7952 bytes
       6/27/99 5:00:00 PM
msdtc.exe      c:\winnt\system32\msdtc.exe   412     8       204800
       1413120 10/21/99 9:16:35 AM    1999.05.3236.5 6928 bytes
       8/22/99 11:29:11 PM
svchost.exe    c:\winnt\system32\svchost.exe 520     8       204800
       1413120 10/21/99 9:16:43 AM    5.00.2051.1    7952 bytes
       6/27/99 5:00:00 PM
regsvc.exe     c:\winnt\system32\regsvc.exe  544     8       204800
       1413120 10/21/99 9:16:43 AM    5.00.2055.1    65296 bytes
       6/27/99 5:00:00 PM
mstask.exe     c:\winnt\system32\mstask.exe  560     8       204800
       1413120 10/21/99 9:16:43 AM    4.71.2064.1    114960 bytes
       8/23/99 6:33:12 AM
explorer.exe   c:\winnt\explorer.exe  628     8       204800  1413120
       10/21/99 9:29:00 AM    5.00.2919.800  236816 bytes   6/27/99
5:00:00 PM
rhostsvr.exe   c:\rhost\rhostsvr.exe  744     8       204800  1413120
       10/21/99 9:29:04 AM    Not Available  111616 bytes   8/23/99
7:35:41 AM
dllhost.exe    Not Available  724     8       Not Available  Not Available
       10/21/99 10:19:47 AM   Unknown Unknown Unknown
msinfo32.exe   c:\program files\common files\microsoft
shared\msinfo\msinfo32.exe    988     8       204800  1413120 10/21/99
5:05:33 PM     5.00.2051.1    16144 bytes    8/23/99 6:33:21 AM
winmgmt.exe    c:\winnt\system32\wbem\winmgmt.exe    1244    8       204800
       1413120 10/21/99 5:05:33 PM    1.50.996.0000  143432 bytes
       6/27/99 5:00:00 PM
rsvp.exe       c:\winnt\system32\rsvp.exe    1320    8       204800
       1413120 10/21/99 5:05:59 PM    5.00.2061.1    162064 bytes
       6/27/99 5:00:00 PM

[Loaded Modules]

Name    Version Size    File Date      Manufacturer   Path
traffic.dll    5.00.2051.1    31504 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\traffic.dll
rsvp.exe       5.00.2061.1    162064 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\rsvp.exe
psapi.dll      5.00.2051.1    28944 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\psapi.dll
rapilib.dll    5.00.2061.1    26384 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\rapilib.dll
rsvpsp.dll     5.00.2061.1    75024 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\rsvpsp.dll
provthrd.dll   1.50.996.0000  69708 bytes    8/23/99 6:33:11 AM
       Microsoft Corporation  c:\winnt\system32\wbem\provthrd.dll
ntevt.dll      1.50.996.0001  180289 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wbem\ntevt.dll
wininet.dll    5.00.2919.800  460560 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wininet.dll
ntmarta.dll    5.00.2063.1    100624 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\ntmarta.dll
perfos.dll     5.0    21776 bytes    6/27/99 5:00:00 PM     Microsoft
Corporation    c:\winnt\system32\perfos.dll
framedyn.dll   1.50.996.0000  159815 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wbem\framedyn.dll
cimwin32.dll   1.50.996.0000  1040463 bytes  6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wbem\cimwin32.dll
wbemess.dll    1.50.996.0000  299071 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wbem\wbemess.dll
mofd.dll       1.50.996.0000  135231 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wbem\mofd.dll
wbemcore.dll   1.50.996.0002  512070 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wbem\wbemcore.dll
winmgmt.exe    1.50.996.0000  143432 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wbem\winmgmt.exe
fastprox.dll   1.50.996.0000  135243 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wbem\fastprox.dll
wbemsvc.dll    1.50.996.0000  135238 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wbem\wbemsvc.dll
wbemcomn.dll   1.50.996.0000  630849 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wbem\wbemcomn.dll
wbemprox.dll   1.50.996.0000  41035 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wbem\wbemprox.dll
cabinet.dll    5.00.2051.1    56080 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\cabinet.dll
msinfo32.dll   5.00.2054.1    307984 bytes   8/23/99 6:33:21 AM
       Microsoft Corporation  c:\program files\common files\microsoft
shared\msinfo\msinfo32.dll
mfc42u.dll     6.00.8447.0    995384 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\mfc42u.dll
msinfo32.exe   5.00.2051.1    16144 bytes    8/23/99 6:33:21 AM
       Microsoft Corporation  c:\program files\common files\microsoft
shared\msinfo\msinfo32.exe
rhostsvr.exe   Not Available  111616 bytes   8/23/99 7:35:41 AM     Not
Available      c:\rhost\rhostsvr.exe
ntshrui.dll    5.00.2051.1    38672 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\ntshrui.dll
linkinfo.dll   5.00.2058.1    16144 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\linkinfo.dll
hhsetup.dll    4.74.8513      67856 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\hhsetup.dll
msvcp50.dll    5.00.7051      565760 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\msvcp50.dll
mmcshext.dll   5.00.2064.1    24336 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\mmcshext.dll
msi.dll 1.02.0527.5    1665808 bytes  6/27/99 5:00:00 PM     Microsoft
Corporation    c:\winnt\system32\msi.dll
powrprof.dll   5.00.2918.2600 13584 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\powrprof.dll
batmeter.dll   5.00.2918.2600 20240 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\batmeter.dll
stobject.dll   5.00.2066.1    83216 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\stobject.dll
webcheck.dll   5.00.2919.800  256784 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\webcheck.dll
netui1.dll     5.00.2053.1    215312 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\netui1.dll
netui0.dll     5.00.2053.1    71952 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\netui0.dll
ntlanman.dll   5.00.2051.1    36112 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\ntlanman.dll
```

```
browseui.dll   5.00.2919.800  808208 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\browseui.dll
shdocvw.dll    5.00.2919.800  1103120 bytes  6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\shdocvw.dll
explorer.exe   5.00.2919.800  236816 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\explorer.exe
msidle.dll     5.00.2919.800  6416 bytes     6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\msidle.dll
mstask.exe     4.71.2064.1    114960 bytes   8/23/99 6:33:12 AM
       Microsoft Corporation  c:\winnt\system32\mstask.exe
regsvc.exe     5.00.2055.1    65296 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\regsvc.exe
wmi.dll 5.00.2067.1    6416 bytes     6/27/99 5:00:00 PM    Microsoft
Corporation    c:\winnt\system32\wmi.dll
netshell.dll   5.00.2066.1    406800 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\netshell.dll
netman.dll     5.00.2059.1    91920 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\netman.dll
sens.dll       5.00.2055.1    36624 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\sens.dll
es.dll  1999.05.3236.5 214800 bytes   6/27/99 5:00:00 PM    Microsoft
Corporation    c:\winnt\system32\es.dll
mtxoci.dll     1999.05.3236.5 103696 bytes   8/22/99 11:29:13 PM
       Microsoft Corporation  c:\winnt\system32\mtxoci.dll
resutils.dll   5.00.2054.1    39184 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\resutils.dll
clusapi.dll    5.00.2054.1    48912 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\clusapi.dll
xolehlp.dll    1999.05.3236.5 19216 bytes    8/22/99 11:29:12 PM
       Microsoft Corporation  c:\winnt\system32\xolehlp.dll
msdtclog.dll   1999.05.3236.5 87312 bytes    8/22/99 11:29:11 PM
       Microsoft Corporation  c:\winnt\system32\msdtclog.dll
mtxclu.dll     1999.05.3236.5 50960 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\mtxclu.dll
msdtcprx.dll   1999.05.3237.2 598288 bytes   8/22/99 11:29:12 PM
       Microsoft Corporation  c:\winnt\system32\msdtcprx.dll
msvcirt.dll    6.10.8383.0    77904 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\msvcirt.dll
txfaux.dll     1999.05.3236.5 376592 bytes   8/22/99 11:29:11 PM
       Microsoft Corporation  c:\winnt\system32\txfaux.dll
msdtctm.dll    1999.05.3236.5 1118480 bytes  8/22/99 11:29:12 PM
       Microsoft Corporation  c:\winnt\system32\msdtctm.dll
msdtc.exe      1999.05.3236.5 6928 bytes     8/22/99 11:29:11 PM
       Microsoft Corporation  c:\winnt\system32\msdtc.exe
wshnetbs.dll   5.00.2051.1    7952 bytes     6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wshnetbs.dll
rasadhlp.dll   5.00.2065.1    7440 bytes     6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\rasadhlp.dll
winrnr.dll     5.00.2064.1    19216 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\winrnr.dll
dhcpcsvc.dll   5.00.2069.1    89872 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\dhcpcsvc.dll
tapi32.dll     5.00.2056.1    125200 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\tapi32.dll
rasman.dll     5.00.2065.1    59664 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\rasman.dll
rasapi32.dll   5.00.2065.1    183568 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\rasapi32.dll
icmp.dll       5.00.2051.1    7440 bytes     6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\icmp.dll
iphlpapi.dll   5.00.2064.2    68880 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\iphlpapi.dll
rnr20.dll      5.00.2066.1    34064 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\rnr20.dll
wshtcpip.dll   5.00.2053.1    17680 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wshtcpip.dll
msafd.dll      5.00.2066.1    52496 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\msafd.dll
rpcss.dll      5.00.2064.1    228624 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\rpcss.dll
svchost.exe    5.00.2051.1    7952 bytes     6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\svchost.exe
iissuba.dll    5.00.0962      10000 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\iissuba.dll
scecli.dll     5.00.2066.1    98064 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\scecli.dll
esent.dll      6.0.3915.0     1020688 bytes  6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\esent.dll
mswsock.dll    5.00.2066.1    63248 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\mswsock.dll
ntdsatq.dll    5.00.2056.1    30992 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\ntdsatq.dll
ntdsa.dll      5.00.2069.1    989456 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\ntdsa.dll
kdcsvc.dll     5.00.2058.1    136976 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\kdcsvc.dll
sfmapi.dll     5.00.2051.1    39696 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\sfmapi.dll
rtutils.dll    5.00.2065.1    44304 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\rtutils.dll
adsldpc.dll    5.00.2054.1    122128 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\adsldpc.dll
activeds.dll   5.00.2054.1    170256 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\activeds.dll
mprapi.dll     5.00.2072.1    91920 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\mprapi.dll
rassfm.dll     5.00.2065.1    21776 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\rassfm.dll
mpr.dll 5.00.2058.1    54544 bytes    6/27/99 5:00:00 PM    Microsoft
Corporation    c:\winnt\system32\mpr.dll
schannel.dll   5.00.2056.1    140048 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\schannel.dll
netlogon.dll   5.00.2069.1    348432 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\netlogon.dll
kerberos.dll   5.00.2063.1    187152 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\kerberos.dll
msprivs.dll    5.00.2056.1    42496 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\msprivs.dll
samsrv.dll     5.00.2064.1    350992 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\samsrv.dll
cryptdll.dll   5.00.2056.1    41232 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\cryptdll.dll
lsasrv.dll     5.00.2065.1    481552 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\lsasrv.dll
lsass.exe      5.00.2065.1    33552 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\lsass.exe
```

```
ntlsapi.dll    5.00.2051.1    6928 bytes     6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\ntlsapi.dll
xactsrv.dll    5.00.2051.1    92432 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\xactsrv.dll
msv1_0.dll     5.00.2066.1    94992 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\msv1_0.dll
wmicore.dll    5.00.2051.1    68880 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wmicore.dll
browser.dll    5.00.2054.1    48400 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\browser.dll
seclogon.dll   5.00.2051.1    17168 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\seclogon.dll
psbase.dll     5.00.2051.1    113424 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\psbase.dll
cryptsvc.dll   5.00.2051.1    67344 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\cryptsvc.dll
wkssvc.dll     5.00.2056.1    88336 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wkssvc.dll
srvsvc.dll     5.00.2053.1    80144 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\srvsvc.dll
cfgmgr32.dll   5.00.2071.1    17168 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\cfgmgr32.dll
dmserver.dll   2068.1.276.12  12048 bytes    6/27/99 5:00:00 PM
       VERITAS Software Corp. c:\winnt\system32\dmserver.dll
winsta.dll     5.00.2054.1    37136 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\winsta.dll
eventlog.dll   5.00.2067.1    44304 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\eventlog.dll
ntdsapi.dll    5.00.2064.1    62224 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\ntdsapi.dll
scesrv.dll     5.00.2052.1    222992 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\scesrv.dll
umpnpmgr.dll   5.00.2069.1    118544 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\umpnpmgr.dll
services.exe   5.00.2063.1    87824 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\services.exe
clbcatq.dll    1999.05.3237.1 496400 bytes   8/22/99 11:29:03 PM
       Microsoft Corporation  c:\winnt\system32\clbcatq.dll
oleaut32.dll   2.40.4501      606480 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\oleaut32.dll
cscui.dll      5.00.2058.1    232720 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\cscui.dll
atl.dll 3.00.8449      58938 bytes    6/27/99 5:00:00 PM     Microsoft
Corporation    c:\winnt\system32\atl.dll
certcli.dll    5.00.2058.1    132880 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\certcli.dll
winspool.drv   5.00.2051.1    110352 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\winspool.drv
winscard.dll   5.00.2053.1    79632 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\winscard.dll
wlnotify.dll   5.00.2056.1    54032 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wlnotify.dll
cscdll.dll     5.00.2069.1    99600 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\cscdll.dll
setupapi.dll   5.00.2071.1    545552 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\setupapi.dll
winmm.dll      5.00.2063.1    186640 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\winmm.dll
lz32.dll       5.00.2052.1    10000 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\lz32.dll
version.dll    5.00.2063.1    16144 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\version.dll
rsabase.dll    5.00.2066.1    129296 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\rsabase.dll
mscat32.dll    5.131.2051.1   7952 bytes     6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\mscat32.dll
imagehlp.dll   5.00.2072.1    40720 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\imagehlp.dll
wintrust.dll   5.131.2051.1   164624 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wintrust.dll
comctl32.dll   5.81    550160 bytes   6/27/99 5:00:00 PM     Microsoft
Corporation    c:\winnt\system32\comctl32.dll
shlwapi.dll    5.00.2919.800  285456 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\shlwapi.dll
shell32.dll    5.00.2919.800  2325264 bytes  6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\shell32.dll
msgina.dll     5.00.2065.1    311568 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\msgina.dll
wsock32.dll    5.00.2066.1    21776 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wsock32.dll
dnsapi.dll     5.00.2058.1    132368 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\dnsapi.dll
wldap32.dll    5.00.2068.1    154896 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\wldap32.dll
ws2help.dll    5.00.2066.1    18192 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\ws2help.dll
ws2_32.dll     5.00.2066.1    68880 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\ws2_32.dll
samlib.dll     5.00.2066.1    46864 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\samlib.dll
netrap.dll     5.00.2051.1    12560 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\netrap.dll
secur32.dll    5.00.2068.1    44816 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\secur32.dll
netapi32.dll   5.00.2066.1    292112 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\netapi32.dll
profmap.dll    5.00.2069.1    27920 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\profmap.dll
ole32.dll      5.00.2064.1    976144 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\ole32.dll
sfc.dll 5.00.2072.1    80144 bytes    6/27/99 5:00:00 PM     Microsoft
Corporation    c:\winnt\system32\sfc.dll
msasn1.dll     5.00.2056.1    50960 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\msasn1.dll
crypt32.dll    5.131.2066.1   456464 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\crypt32.dll
nddeapi.dll    5.00.2051.1    15632 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\nddeapi.dll
userenv.dll    5.00.2062.1    336656 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\userenv.dll
user32.dll     5.00.2059.1    391952 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\user32.dll
gdi32.dll      5.00.2064.1    231696 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\gdi32.dll
rpcrt4.dll     5.00.2071.1    446736 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\rpcrt4.dll
```

```
advapi32.dll   5.00.2067.1    340240 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\advapi32.dll
kernel32.dll   5.00.2063.1    722704 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\kernel32.dll
msvcrt.dll     6.10.8482.0    294965 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\msvcrt.dll
winlogon.exe   5.00.2067.1    158992 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\winlogon.exe
sfcfiles.dll   5.00.2072.1    553232 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\sfcfiles.dll
ntdll.dll      5.00.2064.1    475920 bytes   6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\ntdll.dll
smss.exe       5.00.2054.1    42768 bytes    6/27/99 5:00:00 PM
       Microsoft Corporation  c:\winnt\system32\smss.exe

[Services]

Display Name   Name    State   Start Mode     Service Type   Path    Error
Control Start Name     Tag ID
Alerter Alerter Stopped Disabled       Share Process
       c:\winnt\system32\services.exe        Normal  LocalSystem    0
Application Management AppMgmt Stopped Manual  Share Process
       c:\winnt\system32\services.exe        Normal  LocalSystem    0
Computer Browser       Browser Running Auto    Share Process
       c:\winnt\system32\services.exe        Normal  LocalSystem    0
Indexing Service       cisvc   Stopped Manual  Share Process
       c:\winnt\system32\cisvc.exe   Normal  LocalSystem    0
ClipBook       ClipSrv Stopped Manual  Own Process
       c:\winnt\system32\clipsrv.exe Normal  LocalSystem    0
Distributed File System       Dfs     Stopped Disabled       Own Process
       c:\winnt\system32\dfssvc.exe  Normal  LocalSystem    0
DHCP Client    Dhcp    Stopped Disabled       Share Process
       c:\winnt\system32\services.exe        Normal  LocalSystem    0
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 Disabled       Share Process
       c:\winnt\system32\services.exe        Normal  LocalSystem    0
Event Log      Eventlog       Running Auto    Share Process
       c:\winnt\system32\services.exe        Normal  LocalSystem    0
COM+ Event System      EventSystem    Running Manual  Share Process
       c:\winnt\system32\svchost.exe -k netsvcs     Normal  LocalSystem
       0
Fax Service    Fax     Stopped Manual  Own Process
       c:\winnt\system32\faxsvc.exe  Normal  LocalSystem    0
IIS Admin Service      IISADMIN       Stopped Auto    Share Process
       c:\winnt\system32\inetsrv\inetinfo.exe       Normal  LocalSystem
       0
IMDB Server    ImdbServer     Stopped Disabled       Own Process
       c:\winnt\system32\imdbsrv.exe Normal  LocalSystem    0
Intersite Messaging    IsmServ Stopped Disabled       Own Process
       c:\winnt\system32\ismserv.exe Normal  LocalSystem    0
Kerberos Key Distribution Center      kdc     Stopped Disabled       Share
Process c:\winnt\system32\lsass.exe   Normal  LocalSystem    0
Server  lanmanserver   Running Auto    Share Process
       c:\winnt\system32\services.exe        Normal  LocalSystem    0
Workstation    lanmanworkstation      Running Auto    Share Process
       c:\winnt\system32\services.exe        Normal  LocalSystem    0
License Logging Service       LicenseService Stopped Disabled       Own
Process c:\winnt\system32\llssrv.exe  Normal  LocalSystem    0
TCP/IP NetBIOS Helper Service LmHosts Stopped Disabled       Share Process
       c:\winnt\system32\services.exe        Normal  LocalSystem    0
Messenger      Messenger      Stopped Disabled       Share Process
       c:\winnt\system32\services.exe        Normal  LocalSystem    0
NetMeeting Remote Desktop Sharing     mnmsrvc Stopped Manual  Own Process
       c:\winnt\system32\mnmsrvc.exe Normal  LocalSystem    0
Distributed Transaction Coordinator   MSDTC   Running Auto    Own Process
       c:\winnt\system32\msdtc.exe   Normal  LocalSystem    0
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 Stopped Disabled       Share Process
       c:\winnt\system32\svchost.exe -k netsvcs     Normal  LocalSystem
       0
Plug and Play  PlugPlay       Running Auto    Share Process
       c:\winnt\system32\services.exe        Normal  LocalSystem    0
IPSEC Policy Agent     PolicyAgent    Stopped Disabled       Share Process
       c:\winnt\system32\lsass.exe   Normal  LocalSystem    0
Protected Storage      ProtectedStorage       Running Auto    Share Process
       c:\winnt\system32\services.exe        Normal  LocalSystem    0
Remote Access Auto Connection Manager RasAuto Stopped Manual  Share Process
       c:\winnt\system32\svchost.exe -k netsvcs     Normal  LocalSystem
       0
Remote Access Connection Manager      RasMan  Stopped Manual  Share Process
       c:\winnt\system32\svchost.exe -k netsvcs     Normal  LocalSystem
       0
Routing and Remote Access     RemoteAccess   Stopped Disabled       Share
Process c:\winnt\system32\svchost.exe -k netsvcs     Normal  LocalSystem
       0
Remote Registry Service       RemoteRegistry Running Auto    Own Process
       c:\winnt\system32\regsvc.exe  Normal  LocalSystem    0
Remote Procedure Call (RPC) Locator   RpcLocator     Stopped Manual  Own
Process c:\winnt\system32\locator.exe Normal  LocalSystem    0
Remote Procedure Call (RPC)   RpcSs   Running Auto    Share Process
       c:\winnt\system32\svchost -k rpcss    Normal  LocalSystem    0
QoS 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
```

```
Secondary Logon 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
Print Spooler  Spooler Stopped Disabled       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 Stopped Manual  Share Process
       c:\winnt\system32\svchost.exe -k tapisrv     Normal  LocalSystem
       0
Terminal Services      TermService    Stopped Disabled       Own Process
       c:\winnt\system32\termsrv.exe Normal  LocalSystem    0
Telnet  TlntSvr Stopped Manual  Own Process    c:\winnt\system32\tlntsvr.exe
       Normal  LocalSystem    0
Distributed Link Tracking Server      TrkSvr  Stopped Disabled       Share
Process c:\winnt\system32\services.exe        Normal  LocalSystem    0
Distributed Link Tracking Client      TrkWks  Stopped Disabled       Share
Process c:\winnt\system32\services.exe        Normal  LocalSystem    0
Uninterruptable Power Supply  UPS     Stopped Manual  Own Process
       c:\winnt\system32\ups.exe     Normal  LocalSystem    0
Utility Manager        UtilMan Stopped Manual  Own Process
       c:\winnt\system32\utilman.exe Normal  LocalSystem    0
Windows Time   W32Time Stopped Manual  Share Process
       c:\winnt\system32\services.exe        Normal  LocalSystem    0
World Wide Web Publishing Service     W3SVC   Stopped Manual  Share Process
       c:\winnt\system32\inetsrv\inetinfo.exe       Normal  LocalSystem
       0
Windows Management Instrumentation    WinMgmt Running Manual  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\Communications    All Users:Accessories\Communications  All
Users
Accessories\Entertainment     All Users:Accessories\Entertainment   All
Users
Accessories\System Tools      All Users:Accessories\System Tools    All
Users
Administrative Tools   All Users:Administrative Tools       All Users
Microsoft SQL Server 7.0      All Users:Microsoft SQL Server 7.0    All
Users
Startup All Users:Startup     All Users
Accessories    CLIENT1\Administrator:Accessories    CLIENT1\Administrator
Accessories\Accessibility
       CLIENT1\Administrator:Accessories\Accessibility
       CLIENT1\Administrator
Accessories\Entertainment
       CLIENT1\Administrator:Accessories\Entertainment
       CLIENT1\Administrator
Accessories\System Tools      CLIENT1\Administrator:Accessories\System
Tools   CLIENT1\Administrator
Startup CLIENT1\Administrator:Startup CLIENT1\Administrator

[Startup Programs]

Program Command User Name      Location
Start rhostsvr c:\rhost\rhostsvr.exe -d\rhost -crhosthdlr.exe
       CLIENT1\Administrator  Startup

[OLE Registration]

Object  Local Server
Sound (OLE2)   sndrec32.exe
Media Clip     mplay32.exe
Video Clip     mplay32.exe /avi
MIDI Sequence  mplay32.exe /mid
Sound   Not Available
Media Clip     Not Available
Image Document "C:\Program Files\Windows
NT\Accessories\ImageVue\KodakImg.exe"
WordPad Document       "%ProgramFiles%\Windows NT\Accessories\WORDPAD.EXE"
Bitmap Image   mspaint.exe
```

Component Services Configuration

```
Component Services Configuration

COM+ Component TPCC.AllTxns Settings:
   Transactions not supported
   Enable object pooling
   Minimum pool size 76
   Maximum pool size 76
   Creation timeout 60,000
   Enable object construction
   Enable just in time activation
   Concurrency required
```

Internet Information Server Registry Parameters

```
\registry\machine\system\currentcontrolset\services\inetinfo
    Parameters
        ListenBackLog = REG_DWORD 0x00000019
        DispatchEntries = REG_MULTI_SZ "LDAPSVC"
        PoolThreadLimit = REG_DWORD 0x000000a0
        ThreadTimeout = REG_DWORD 0x00015180
        BandwidthLevel = REG_DWORD 0xffffffff
```

```
        DisableMemoryCache = REG_DWORD 0x00000001
        MemoryCacheSize = REG_DWORD 0x00000000
        ObjectCacheTTL = REG_DWORD 0xffffffff
    Performance
        Library = infoctrs.dll
        Open = OpenINFOPerformanceData
        Close = CloseINFOPerformanceData
        Collect = CollectINFOPerformanceData
        Last Counter = REG_DWORD 0x00000818
        Last Help = REG_DWORD 0x00000819
        First Counter = REG_DWORD 0x000007d8
        First Help = REG_DWORD 0x000007d9
```

World Wide Web Server Registry Parameters

```
\registry\machine\system\currentcontrolset\services\w3svc [17 1]
    Type = REG_DWORD 0x00000020
    Start = REG_DWORD 0x00000003
    ErrorControl = REG_DWORD 0x00000001
    ImagePath = REG_EXPAND_SZ C:\WINNT\System32\inetsrv\inetinfo.exe
    DisplayName = World Wide Web Publishing Service
    DependOnService = REG_MULTI_SZ "IISADMIN"
    DependOnGroup = REG_MULTI_SZ
    ObjectName = LocalSystem
    Description = Provides Web connectivity and administration through the
Internet Information Services snap-in.
    ASP [17 1]
        NOTE = This is for backward compatibility only.
        Parameters [17 1]
    Parameters [17 1]
        MajorVersion = REG_DWORD 0x00000005
        MinorVersion = REG_DWORD 0x00000000
        InstallPath = C:\WINNT\System32\inetsrv
        CertMapList = C:\WINNT\System32\inetsrv\iiscrmap.dll
        AccessDeniedMessage = Error: Access is Denied.
        Filter DLLs =
        AcceptExOutstanding = REG_DWORD 0x00000028
        ADCLaunch [17 1]
            AdvancedDataFactory [17 1]
            RDSServer.DataFactory [17 1]
        Script Map [17 1]
        Virtual Roots [17 1]
            / = c:\inetpub\wwwroot,,207
            /IISAdmin = C:\WINNT\System32\inetsrv\iisadmin,,201
            /IISSamples = c:\inetpub\iissamples,,201
            /MSADC = c:\program files\common files\system\msadc,,205
            /IISHelp = c:\winnt\help\iishelp,,201
            /Scripts = c:\inetpub\scripts,,204
            /Printers = C:\WINNT\web\printers,,201
    Performance [17 1]
        Library = w3ctrs.dll
        Open = OpenW3PerformanceData
        Close = CloseW3PerformanceData
        Collect = CollectW3PerformanceData
        Last Counter = REG_DWORD 0x000008c8
        Last Help = REG_DWORD 0x000008c9
        First Counter = REG_DWORD 0x00000826
        First Help = REG_DWORD 0x00000827
    Security [17 1]
        Security = REG_BINARY 0x000000b8 0x80140001 0x000000a0 0x000000ac
0x00000014 0x00000030 0x001c0002 0x00000001 0x00148002 0x000f01ff
0x00000101 0x01000000 0x00000000 0x00700002 0x00000004 0x00180000
0x000201fd 0x00000101 0x05000000 \
                  0x00000012 0x006f0074 0x001c0000 0x000f01ff 0x00000201
0x05000000 0x00000020 0x00000220 0x00730072 0x00180000 0x0002018d
0x00000101 0x05000000 0x0000000b 0x00000220 0x001c0000 0x000201fd
0x00000201 0x05000000 0x00000020 \
                  0x00000223 0x00730072 0x00000101 0x05000000 0x00000012
0x00000101 0x05000000 0x00000012
    Enum [17 1]
        0 = Root\LEGACY_W3SVC\0000
        Count = REG_DWORD 0x00000001
        NextInstance = REG_DWORD 0x00000001
```

TPCC Application Settings

```
\registry\machine\software\unisys
    TPCC
        MAXTERMS = 10000
        SERVERNAME = AVALON4
        DELIVERYTHREADS = 9
        DATABASE = tpcc
```

Appendix D - RTE Code

Admin Environment

```
if '%1'==' goto usage
if '%2'==' goto usage
if '%3'==' goto usage

:paramok

set WEBCHECKWIDS=1
set WEBDIAGLEVEL=4
set WEBEVENTLOG=0
set WEBEVENTHOST=
set WEBCHECKLEVEL=2

c:\webdriver\webadmin.exe -cweb%1.cfg -m%2 -d%3 -s160
if %ERRORLEVEL% NEQ 0 pause

goto end

:usage
@ECHO You must supply the following parameters:
@ECHO "webnnn.cmd <cfg file suffix> <min driver #> <max driver #>"
pause

:end
```

Profiles used for Performance Run

Web2148.cfg

```
//
//  Common Driver Configuration
//
INITBASEPORT 4300
INITSYNCMAX 4
INITPAUSE 1
INITRSCALE 400
INITTSCALE 100
INITRWID 1, 2148
INITFIXEDWID 1
INITCCLAST 208
INITCCID 208
INITCITEMID 208
//
//  Configuration Driver 1
//
1 INITIPADDR 192.168.90.31
1 INITIISADDR 192.168.10.1
1 INITIISPORT 80
1 INITBROWSERS 900
1 INITMYWID 1,90

//
//  Configuration Driver 2
//
2 INITIPADDR 192.168.90.31
2 INITIISADDR 192.168.20.2
2 INITIISPORT 80
2 INITBROWSERS 900
2 INITMYWID 91,180

//
//  Configuration Driver 3
//
3 INITIPADDR 192.168.90.31
3 INITIISADDR 192.168.30.3
3 INITIISPORT 80
3 INITBROWSERS 900
3 INITMYWID 181,270

//
//  Configuration Driver 4
//
4 INITIPADDR 192.168.90.32
4 INITIISADDR 192.168.11.1
4 INITIISPORT 80
4 INITBROWSERS 900
4 INITMYWID 271,360

//
//  Configuration Driver 5
//
5 INITIPADDR 192.168.90.32
5 INITIISADDR 192.168.21.2
5 INITIISPORT 80
5 INITBROWSERS 900
5 INITMYWID 361,450

//
//  Configuration Driver 6
//
6 INITIPADDR 192.168.90.32
6 INITIISADDR 192.168.31.3
6 INITIISPORT 80
6 INITBROWSERS 900
6 INITMYWID 451,540

//
//  Configuration Driver 7
//
7 INITIPADDR 192.168.90.33
7 INITIISADDR 192.168.12.1
```

```
7 INITIISPORT 80
7 INITBROWSERS 900
7 INITMYWID 541,630

//
//  Configuration Driver 8
//
8 INITIPADDR 192.168.90.33
8 INITIISADDR 192.168.22.2
8 INITIISPORT 80
8 INITBROWSERS 900
8 INITMYWID 631,720

//
//  Configuration Driver 9
//
9 INITIPADDR 192.168.90.33
9 INITIISADDR 192.168.32.3
9 INITIISPORT 80
9 INITBROWSERS 900
9 INITMYWID 721,810

//
//  Configuration Driver 10
//
10 INITIPADDR 192.168.90.34
10 INITIISADDR 192.168.13.1
10 INITIISPORT 80
10 INITBROWSERS 900
10 INITMYWID 811,900

//
//  Configuration Driver 11
//
11 INITIPADDR 192.168.90.34
11 INITIISADDR 192.168.23.2
11 INITIISPORT 80
11 INITBROWSERS 900
11 INITMYWID 901,990

//
//  Configuration Driver 12
//
12 INITIPADDR 192.168.90.34
12 INITIISADDR 192.168.33.3
12 INITIISPORT 80
12 INITBROWSERS 900
12 INITMYWID 991,1080

//
//  Configuration Driver 13
//
13 INITIPADDR 192.168.90.35
13 INITIISADDR 192.168.14.1
13 INITIISPORT 80
13 INITBROWSERS 890
13 INITMYWID 1081,1169

//
//  Configuration Driver 14
//
14 INITIPADDR 192.168.90.35
14 INITIISADDR 192.168.24.2
14 INITIISPORT 80
14 INITBROWSERS 890
14 INITMYWID 1170,1258

//
//  Configuration Driver 15
//
15 INITIPADDR 192.168.90.35
15 INITIISADDR 192.168.34.3
15 INITIISPORT 80
15 INITBROWSERS 890
15 INITMYWID 1259,1347

//
//  Configuration Driver 16
//
16 INITIPADDR 192.168.90.36
16 INITIISADDR 192.168.15.1
16 INITIISPORT 80
16 INITBROWSERS 890
16 INITMYWID 1348,1436

//
//  Configuration Driver 17
//
17 INITIPADDR 192.168.90.36
17 INITIISADDR 192.168.25.2
17 INITIISPORT 80
17 INITBROWSERS 890
17 INITMYWID 1437,1525

//
//  Configuration Driver 18
//
18 INITIPADDR 192.168.90.36
18 INITIISADDR 192.168.35.3
18 INITIISPORT 80
18 INITBROWSERS 890
18 INITMYWID 1526,1614

//
//  Configuration Driver 19
//
19 INITIPADDR 192.168.90.37
19 INITIISADDR 192.168.16.1
19 INITIISPORT 80
19 INITBROWSERS 890
19 INITMYWID 1615,1703

//
//  Configuration Driver 20
//
20 INITIPADDR 192.168.90.37
20 INITIISADDR 192.168.26.2
20 INITIISPORT 80
```

```
20 INITBROWSERS 890
20 INITMYWID 1704,1792

//
//  Configuration Driver 21
//
21 INITIPADDR 192.168.90.37
21 INITIISADDR 192.168.36.3
21 INITIISPORT 80
21 INITBROWSERS 890
21 INITMYWID 1793,1881

//
//  Configuration Driver 22
//
22 INITIPADDR 192.168.90.38
22 INITIISADDR 192.168.17.1
22 INITIISPORT 80
22 INITBROWSERS 890
22 INITMYWID 1882,1970

//
//  Configuration Driver 23
//
23 INITIPADDR 192.168.90.38
23 INITIISADDR 192.168.27.2
23 INITIISPORT 80
23 INITBROWSERS 890
23 INITMYWID 1971,2059

//
//  Configuration Driver 24
//
24 INITIPADDR 192.168.90.38
24 INITIISADDR 192.168.37.3
24 INITIISPORT 80
24 INITBROWSERS 890
24 INITMYWID 2060,2148

//
```

Driver Environment

```
if '%1'==' goto usage

:paramok

set WEBDRIVERNO=%1
set WEBADMBASEPORT=4300
set WEBDIAGLEVEL=2
set WEBEVENTLOG=1
set WEBEVENTHOST=
set WEBLOGLEVEL=1
set WEBSINGLETRAN=0
set WEBTPCCAUDIT=0
set WEBRTFUDGETM=110
set WEBNEWORDERPROB=4484
set WEBPAYMENTPROB=4307
set WEBORDERSTATUSPROB=403
set WEBDELIVERYPROB=403
set WEBSTOCKLEVELPROB=403
set WEBTTNEWORDER=12030
set WEBTTPAYMENT=12030
set WEBTTDELIVERY=5060
set WEBTTORDERSTATUS=10070
set WEBTTSTOCKLEVEL=5060

webdriver.exe

goto end

:usage
@ECHO  You must supply the following parameters:
@ECHO  "webdriver.cmd <driver number>"
pause

:end
exit
```

Appendix E - Disk Storage

TPC-C 180-Day Disk Space Requirements

Warehouses	**2160**	tpmC	**26,922.60**	tpmC/W	12.46

Table	**Initial Rows**	**Data KB**	**Index KB**	**Extra 5% KB**	**Total With 5% KB**
Warehouse	2,160	232	40	14	286
District	21,600	2,400	64	123	2,587
Customer	64,800,000	47,127,280	3,026,176	2,507,673	52,661,129
History (D)	64,800,000	3,600,064	0		3,600,064
Order (D)	64,800,000	1,986,208	1,217,120		3,203,328
New-Order	19,440,000	307,352	848	15,410	323,610
Order-Line (D)	647,996,638	40,499,792	100,872		40,600,664
Item	100,000	9,528	88	481	10,097
Stock	216,000,000	69,120,000	154,928	3,463,746	72,738,674
Totals KB		162,652,856	4,500,136	5,987,447	173,140,439

Db/Filegroup	**Count**	**Size MB**	**MB Allocated**	**MB Loaded +5%**	**MB for 8 Hours**
master, model & msdb		22	22	22	22
tempdb		10	10	10	10
mssql70_tpcc_root	1	10	10	10	10
mssql70_cs_fg	7	18,000	126,000	122,461	122,461
mssql70_misc_fg	7	9,500	66,500	46,622	11,172
Total Allocated MB			**192,542**	**169,124**	**133,675**

	MB	
Dynamic Space MB	45,006	Sum of data for orders, order_line & history
Static Space	124,077	Sum of data+index+5% - Dynamic Space
Free Space	23,460	Total allocated space - (Dynamic & Static Spaces)
Daily Growth	8,975	(Dynamic Space / (W * 62.5)) * tpmC
Daily Spread	9,996	Free space - 1.5 * Daily growth (zero if negative)
	0	SQL Server can be configured to eliminate Daily Spread
180 Day Space MB	1,739,645	Static Space + 180 * (Daily Growth + Daily Spread)
180 Day Space GB	**1,698.87**	
8 hr log GB	**65.07**	(need double for mirroring)

Disk Capacity MB	4372	**4.2695 GB**	Capacity of 4GB disks
	8747	**8.5420 GB**	Capacity of 9GB disks
	17496	**17.0859 GB**	Capacity of 18GB disks

Space Usage	**GB Needed**	**Disks Priced**	**GB Priced**	
180-day space DB	1698.87 GB	0	0.00 GB	18GB drives
		224	1913.41 GB	9GB drives
Total DB		**224**	1913.41 GB	
8-hr log+mirror	130.13 GB	16	136.67 GB	9GB drives
OS, SQL Server	4.24 GB	1	8.50 GB	9GB drive
Total space	**1833.24 GB**	**241**	**2058.58 GB**	

TPC-C 180-Day Dynamic Table Growth Rates for 8 Hours						**26,922.60 tpmC**	
Tables	**Initial (KB)**	**Final (KB)**	**Change(KB)**	**Unused (KB)**	**KB / New-Order**	**8-Hr MB**	
History	3,600,064	4,392,232	792,168	35,712	0.0636	803.19	
Orders	3,203,328	4,925,936	1,722,608	2,856	0.1384	1,746.58	
Order_line	40,600,664	48,780,144	8,179,480	6,584	0.6572	8,293.33	
Dynamic	**47,404,056**	**58,098,312**	**10,694,256**	**45,152**	**0.8592**	**10,843.10**	
New_order	308,200	517,888	209,688	22,696	0.0168	212.61	
Static							
Log	523,364	65,732,629	65,209,265		**5.2391**	66,627.97	65.066
SUM(d_next_o_id)	64,821,600	77,268,330	12,446,730				

Appendix F - Third-Party Price Quotations

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
Tel 425 882 8080
Fax 425 936 7329
http://www.microsoft.com/

Microsoft

October 4, 1999

Mr. Jerrold Buggert
Director, Systems Analysis, Modeling, Measurement
Unisys Corporation
25725 Jeronimo Road
Mission Viejo, CA 92691
949-380-5106
Fax (949) 465-2552

Dear Mr. Buggert:

Here is the information you requested regarding U.S. pricing of several Microsoft products to be used in conjunction with TPC-C benchmark testing.

Microsoft SQL Server 7.0, Enterprise Edition (one server w/ unlimited CALs)	**$28,999**
Microsoft Windows NT Server 4.0, Enterprise Edition (one server w/ 25 CALs)	**$3,999**
Windows 2000 Server (one server w/ 25 CALs, no discount for additional servers)	**$809**
Visual C++ Professional 6.0 (single copy)	**$549**
5-year maintenance for above software @ $2095/yr	**$10,475**

This quote is valid for the next 90 days.

Some products may not be currently orderable but will be available through Microsoft's normal distribution by December 31, 1999.

If I can be of any further assistance, please contact me at 425-936-5301 or tomkr@microsoft.com.

Yours truly,

Thomas Kreyche
Product Manager
SQL Server Marketing

Western Micro Technology **10/25/99**
(800)937-8446

Quoted to: Jerry Buggert/Unisys for TPC.org
Prepared by: Tony Jacobs

Qty.	Description	Style	Price	Extended Price
1	SYS: Aquanta ES5085R, w/ CDRom, 0 Proc, 0MB Mem	ESR508151-GZN	$18,196	$18,196
4	PROC: 550MHz Pentium III Xeon /2MB Cache & VRMs	XEO3550-2MB	$5,893	$23,572
1	BRD: Processor Mezzanine Board, 0 Proc.	ESR81-MEZ	$1,179	$1,179
32	MEM: 128 MB Memory, SDRAM, Buf 6ns	DIM6168-128	$442	$14,144
1	BRD: Memory Carrier Board, 0 Mem.	ESR81-MCB	$737	$737
2	MEM: Cache Coherency Filter, 4x SRAM	ESR81-CC4	$921	$1,842
1	DISK: 9GB, 10K SCSI LVD, SCA	HDL91102-CX1	$516	$516
1	ETHERNET: 10/100Mbit/sec, PCI 32-bit	ETH1010052-PCI	$100	$100
1	SYS MGT: ES5080 Value Add Software	ESS508011-N	$368	$368
1	MONITOR: 15-inch Color	EVG2100-P	$221	$221
1	KEYBD: 104 Key Spacesaver	PCK104-SKB	$26	$26
1	MOUSE: 2 Button PS2	PWM1-PS2	$15	$15
	Server Total			**$60,916**
264	DISK: 9GB Drive, 10K SCSI LVD, SCA + 10% spares	OSD9205-W45	$566	$149,424
28	CAB: Disk, 8 SCA w/ I/F cards, 0 Disks, 3U	OSM310300-L05	$2,118	$59,304
28	ACC: Deskside Pedestal	OSM3000-DSK	$31	$868
30	CBL: SCSI 68-pin VHD Conn's, 5 meter	CBL134-5	$142	$4,260
2	CAB: Disk, 8 SCA w/ RAID Cntl'r, 0MB, 0 Disks, 3U	OSM311000-LR	$4,191	$8,382
2	MEM: 32MB OSM cache	OSM1032-MEM	$187	$374
2	PWR: 2nd Power Supply Upgrade, OSM	OSM3000-BPF	$392	$784
1	PWR:3000 VA UPS, 3U	UPD30001-SXR	$1,897	$1,897
3	PWR: Distribution Kit, 220V	SFR220-PWR	$42	$126
3	CAB: Rackmount Kit for Disk Cages	OSM3000-RMK	$97	$291
1	CAB: 36U x 19" x 34" Cabinet, Open	RM361934-OFT	$884	$884
1	DOOR: 36U x 19", Rear	RM3619-RDR	$277	$277
1	PNL: 36U x 34" Side Skins, L&R	RM3634-SDS	$221	$221
	Storage Total			**$227,092**
3	SYS: NetServer LC3, w/1 550MHz Proc., 0MB Mem	D8592-AV	$1,635	$4,905
6	MEM: 64 MB SDRAM Memory Upgrade	D6097-AV	$110	$660
6	MEM: 128 MB SDRAM Memory Upgrade	D6098-AV	$159	$954
3	DISK: 4GB SCSI 3.5 Internal	D4910-AV	$303	$909
6	ETHERNET: 10/100TX Mbit/sec, PCI 32-bit	D5013-AV	$68	$408
3	MONITOR:15-inch Color	EVG2100-P	$221	$663
	Client Total			**$8,499**
	Server, Storage and Client Total			**$296,507**
	Discount based on total dollar volume			**($29,651)**
Quote Total				**$266,856**

Quote valid for 90 days.
Disks come with return to factory, 5 year warranty, 7 day replenishment

American Megatrends

Date: 10/7/99

Contact Name: **Rick Freeman**
Company: **Unisys**

Phone Number: **(949)-380-5539**
Fax Number: **(949)-380-5344**

MegaRAID Enterprise 1500-H, PCI SCSI Disk Array Controller Quotation

Price Quote

MegaRAID Product	Qty 1-15	Description
MegaRAID Enterprise 1500-H	**$2200.00**	**Enterprise 1500-H, P/N 4674536264, 4CH, 64MB SDRAM & BBU**
Product Warranty	**$60.00 Per Unit**	**Extended Warranty for two years**

Distinguishing Features:
⇒ Boot-up Configuration Utility
⇒ AMI high performance RAID Firmware on Flash EPROM
⇒ Support for Low Voltage Differential

Conditions:
- All pricing are quoted FOB factory, Norcross, GA. Shipping and insurance are additional.
- Quotation is subject to the execution of a Purchase Agreement.
- 3 years limited warranty with optional 2 years extended warranty for amount of $60 per unit.
- RMA is in accordance with AMI's standard. Return and Repair within 7 days.
- Product is available now.

Deliverables:
- Products will ship in bulk packaging or individually depending on quantity, each sealed in an anti-static bag.
- Manuals and Driver(s).

Submitted by: Siamak Iranpour
Senior RAID Program Manager

This quotation is valid for 90 days from the date shown and is subject to the conditions as listed.

Software House International **Pricing Proposal**	**Quotation #MO-991007-37695** **10/07/99**

Unisys
Rick Freeman
Quote Good For Ninety Days

Phone: **Fax: 949-465-2552**

SHI Account Exec: Matthew O. Martin
Telephone : (408) 922-1106
Fax : (408) 526-1222

Reference:

Product	Part #	Qty	List	Your Price	Total
8Port 10BT Hub +1 5 Year return to man. warranty Qantity 2800 Plus	Z99552	2800		$27.00	$75,600.00
Total					**$75,600.00**

Additional Comments:

NETLUX

14180 Live Oak Ave., Unit E
Baldwin Park, Ca. 91760

1-800-789-1780
Phone#626-851-9737
Fax #626-851-9837

October 7, 1999

Rick Freeman
Unisys Corporation
25725 Jeronimo Road
Mission Viejo, CA 92691
Fax: (949) 380-5539
cc: (949) 380-5344

Quotation

Quantity	Part No.	Description	Unit Price
1-50	NX-SW8	NETLUX 8-port 10/100Mbps FAST Ethernet Switch	$229.00

Terms and Conditions:
FOB Origin
Quote Valid for 90 days
5 Year Warranty

Sincerely,
Martin Parry
NETLUX