



TPC Benchmark® C

Full Disclosure Report

HP Integrity Superdome
using Microsoft SQL Server 2000 Enterprise Edition 64-bit
on Microsoft Windows Server 2003, Datacenter edition (64-bit)

Second Edition
August 28, 2003

Second Edition - August 28, 2003

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

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

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

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

© Copyright Hewlett-Packard Company 2003.

All rights reserved. Permission is hereby granted to reproduce this document in whole or in part provided the copyright notice printed above is set forth in full text on the title page of each item reproduced.

Printed in U.S.A., August 28, 2003

HP and HP StorageWorks are registered trademarks of Hewlett-Packard Company.

Microsoft Windows NT, SQL Server and COM+ are registered trademarks of Microsoft Corporation.

TPC Benchmark, TPC-C, and tpmC are registered certification marks of the Transaction Processing Performance Council.

All other brand or product names mentioned herein are trademarks or registered trademarks of their respective owners.

Abstract

Overview

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

TPC Benchmark® C Metrics

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

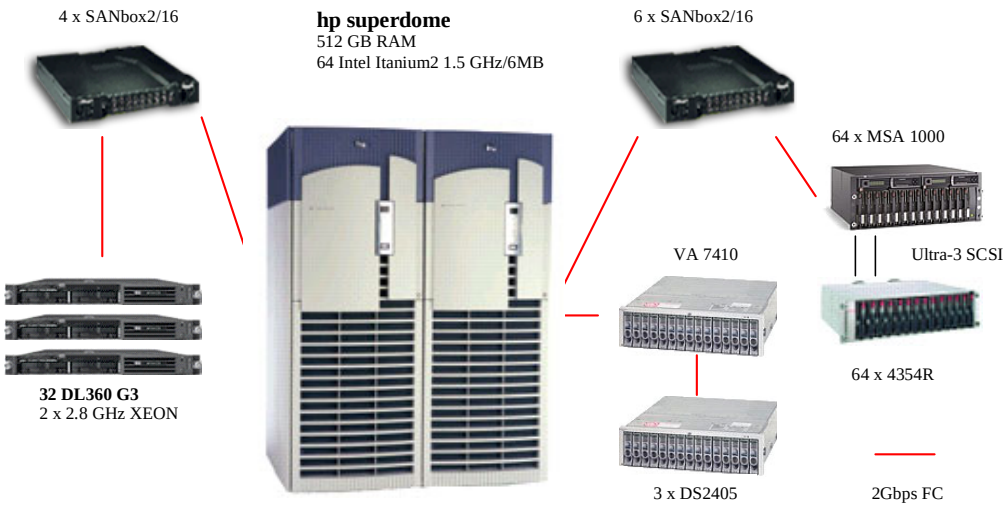
Standard and Executive Summary Statements

The following pages contain the executive summary of the benchmark results for the HP Integrity Superdome system. The Standard System Summary is given below.

Company Name	System Name	Database Software	Operating System
Hewlett-Packard Company	HP Integrity Superdome	Microsoft SQL Server 2000 Enterprise Edition 64-bit	Microsoft Windows Server 2003, Datacenter edition (64-bit)
Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
\$5,105,486	786646.62 tpmC	\$6.49 per tpmC	October 23, 2003

Auditor

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

	HP Integrity Superdome			TPC-C Revision 5.1
Total System Cost	TPC Throughput	Price/Performance	Report Date	Availability Date
\$5,105,486	786646.62 tpmC	\$6.49 per tpmC	August 28, 2003	October 23, 2003
Processors	Database Manager	Operating System	Other Software	Number of Users
64 Intel Itanium2 1.5GHz - Server 64 x Intel Xeon 2.8GHz - Client	Microsoft SQL Server 2000 Enterprise Edition 64-bit	Microsoft Windows Server 2003, Datacenter edition (64-bit)	Microsoft Visual C++ Microsoft COM+ Transaction Monitor	635,920
 The diagram illustrates the HP Integrity Superdome system architecture. At the center is the hp superdome server, which is configured with 512 GB RAM and 64 Intel Itanium2 1.5 GHz/6MB processors. To the left, four SANbox2/16 units are connected to the Superdome. Below these, 32 DL360 G3 servers (2 x 2.8 GHz XEON) are shown. To the right, six SANbox2/16 units are connected to the Superdome. Below these, a VA 7410 storage unit is connected to the Superdome, which in turn connects to three DS2405 storage units. Additionally, 64 x MSA 1000 storage units are connected to the Superdome via Ultra-3 SCSI, which then connects to 64 x 4354R storage units. Finally, 2Gbps FC connections are shown between the 4354R units and the DS2405 units.				
System Components	Server		Each Client	
	Qty	Type	Qty	Type
Processors	64	Intel Itanium 2 6M CPUs at 1.5GHz	2	2.8 GHz Intel Xeon
Cache Memory		6 MB iL3 cache		512kbyte L2 Cache
Memory	128	4 Gbyte	4	256 Mbyte
Disk Controllers	64	HP StorageWorks MSA 1000	1	Embedded Smart Array 5i+
Disk Drives	1792	HP 18GB 15KRPM U320	1	36 Gbyte disk
	60	HP 73GB, 10krpm FC hot-swap disk		
	2	HP 18GB 15KRPM for OS		
Total Storage		34586.4 Gbyte		36 Gbyte
Tape Drives	1	HP Surestore DAT40e		
Terminals	1	Console Terminal	1	Console Terminal

HP Hewlett Packard Company		HP Integrity Superdome			TPC-C Rev. 5.1	
					Report Date: August 28, 2003	
Description	Price Key	Part Numbr	Unit Price	Qty	Extended Price	3 Yr Maint Price
Superdome left cabinet	1	A5201A opt429	\$205,840	1	\$205,840	
Superdome right cabinet	1	A5202A opt429	\$218,435	1	\$218,435	
IPF Superdome Cell Board (sx1000)	1	A6866A	\$16,000	16	\$256,000	
3 Year Svc & Support Price (Hardware and Software)						\$869,480
Itanium 2 6M, 1.5GHz processor module (2 CPUs)	1	A6924A	\$40,000	32	\$1,280,000	
4GB SDRAM (4x1GB DIMMs)	1	A6863A	\$17,750	128	\$2,272,000	
PCI-x I/O chassis	1	A6864A	\$16,805	8	\$134,440	
Core I/O card	1	A6865A	\$1,045	1	\$1,045	
Gig E - TX Adapter	1	A7061A	\$270	1	\$270	
Graphics USB Card	1	A6869A	\$349	1	\$349	
2-port SCSI adapter	1	A7060A	\$600	1	\$600	
FC-HBA 2GB 1 Channel	1	LP9802-F2	\$1,695	2	\$3,390	
FCA2214 FC-HBA for MSA	1	281541-B21	\$1,590	16	\$25,440	
Qlogic QLA-2350 Fibre-Channel VI Adapter	3	QLA2350-BK	\$1,765	17	\$30,005	
5m LC to LC Cable Kit	1	221692-B22	\$82	80	\$6,560	
5m 2gbit Small Form Pluggable Adapter Kit	1	221470-B21	\$369	80	\$29,520	
Modular Storage Array 1000	1	201723-B22	\$9,995	64	\$639,680	
3 Year Support Price						\$236,480
18GB, 15krpm Ultra320 Wide disk (10% spares)	1	286775-B22	\$359	180	\$64,620	
Storageworks enclosure 4354R	1	190211-001	\$3,523	64	\$225,472	
18.2GB, 15krpm Ultra320 Wide disk	1	286775-B22	\$359	1792	\$643,328	
HP9000 Standard Rack System E41	1	A4902A	\$1,910	14	\$26,740	
HP Power Distribution Unit 120-240V	1	A5137AZ	\$145	27	\$3,915	
VA7410 with dual 1 GB cache (for log) via Emulex	1	A6267AZ	\$77,300	1	\$77,300	
3 Year Support Price						\$9,563
DS2405 (add on to VA array)	1	A6250AZ	\$6,595	3	\$19,785	
DS2100 (boot)	1	A5675AZ	\$775	1	\$775	
18 GB 15krpm drive for DS2100	1	A6540A	\$800	2	\$1,600	
HP 73GB, 10krpm FC hot-swap disk	1	A6194A	\$2,019	60	\$121,140	
HP 73GB, 10krpm FC hot-swap disk (spares)	1	A6194A	\$2,019	6	\$12,114	
SCSI Cable 10m VHDS68/HDS68 M/M Multim	1	C2363B	\$335.00	2	\$670	
SCSI Terminator LVD/SE HDS68 Multimd	1	C2364A	\$100.00	1	\$100	
Fbr Optic Cable 2m LC 50/125 M/M	1	C7524A	\$215.00	6	\$1,290	
Fbr Optic Cable 16m LC 50/125 M/M	1	C7525A	\$260.00	2	\$520	
TA5300 Enclosure for DAT tape	1	C7508AZ	\$1,395	1	\$1,395	
DVD Rom drive	1	C7499A	\$688	1	\$688	
DAT Tape	1	C7497A	\$2,155	1	\$2,155	
Proliant ML350 G3 (SMS for Orca)	1	A9802A	\$4,700	1	\$4,700	
Server Subtotal					\$6,311,881	\$1,115,523
Microsoft Windows Server 2003, Datacenter edition (64-bit)	1	T2372A opt064	\$128,000	1	\$128,000	
Microsoft SQL Server 2000 Enterprise Edition 64 bit	2	810-00560	\$16,541	64	\$1,058,624	
Database Server Support Package	2	PRO-PRORS-16U-01	\$1,950	3		\$5,850
Server Software Subtotal					\$1,186,624	\$5,850
HP ProLiant DL360 G3	1	308724-001	\$3,249	32	\$103,968	\$75,360
2.80GHz Xeon Processor Option Kit (DL360 G3)	1	292892-B21	\$999	32	\$31,968	
HP Mouse	1	170299-B22	\$23	1	\$23	
HP Enhanced Keyboard	1	265977-001	\$12	1	\$12	
S5500 15 carbon / silver monitor	1	261602-001	\$129	1	\$129	
36.4-GB 15,000 rpm U320 Universal Hard Drive	1	286776-B22	\$649	32	\$20,768	
Qlogic QLA-2350 Fibre-Channel VI Adapter	3	QLA2350-BK	\$1,765	37	\$65,305	
5M LC to LC Cable Kit	1	221692-B22	\$82	40	\$3,280	
5m 2gbit Small Form Pluggable Adapter Kit	1	221470-B21	\$369	40	\$14,760	
Client Subtotal					\$240,213	\$75,360
Microsoft Windows 2000 Server	2	C11-00821	\$738	32	\$23,616	
Microsoft Visual C++ Standard 6.0	2	254-00170	\$109	1	\$109	\$1,009
Client Software Subtotal					\$23,725	\$1,009
Qlogic SANBox-2 16-Port Switch	3	SANBOX 2/16	\$15,995	12	\$191,940	
Connectivity Subtotal					\$191,940	
HP's Large Configuration Discount *	* Discounts:				-\$3,528,267	-\$518,372
Total:					\$4,426,116	\$679,370
Price Key: 1-HP, 2-Microsoft, 3-Qlogic, 4-CDW Audited by Lorna Livingtree of Performance Metrics, Inc.	3 year cost of ownership:					\$5,105,486
	tpmC:					786,646.62
* All discounts are based on US list prices and for similar quantities and configurations	\$/tpmC:					\$6.49
Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specifications. If you find that the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.						

Numerical Quantities Summary for HP Integrity Superdome

MQTH, Computed Maximum Qualified Throughput

786646.62 tpmC

Response Times (in seconds)

	90th %-ile	Maximum	Average
New-Order	1.33s	56.51s	0.69s
Payment	1.30s	28.21s	0.65s
Order-Status	1.33s	28.29s	0.67s
Delivery (interactive portion)	0.11s	6.51s	0.10s
Delivery (deferred portion)	0.15s	3.13s	0.10s
Stock-Level	1.56s	10.19s	0.87s
Menu	0.11s	13.62s	0.10s

Response time delay added for emulated components 0.1 seconds

Transaction Mix, in percent of total transactions

New-Order	44.96%
Payment	43.01%
Delivery	4.01%
Stock-Level	4.01%
Order-Status	4.01%

Keying/Think Times

	Keying Time			Think Time		
	Min	Avg	Max	Min	Avg	Max
New-Order	18s	18.02s	18.03s	0s	12.05s	120.51s
Payment	3s	3.02s	3.03s	0s	12.04s	120.51s
Order-Status	2s	2.02s	2.03s	0s	10.05s	100.51s
Delivery (interactive)	2s	2.02s	2.03s	0s	5.06s	50.51s
Stock-Level	2s	2.02s	2.03s	0s	5.05s	50.51s

Test Duration

Ramp up time	45 minutes
Measurement interval	120 minutes
Transactions during measurement interval	218374332
Ramp down time	38 minutes

Checkpointing

Number of checkpoints in measurement interval	8
Checkpoint Interval	15 minutes

Table of Contents

Abstract	1
Overview	1
TPC Benchmark® C Metrics.....	1
Standard and Executive Summary Statements.....	1
Auditor	1
Table of Contents	5
Preface	7
Document Structure.....	7
TPC Benchmark® C Overview	7
System Overview	8
General Items	9
Test Sponsor	9
Application Code and Definition Statements.....	9
Parameter Settings	9
Configuration Diagrams.....	9
Chapter 1 Logical Database Design.....	11
1.1 Table Definitions	11
1.2 Physical Organization of the Database.....	11
1.3 Insert and Delete Operations.....	11
1.4 Partitioning.....	11
1.5 Replication, Duplication or Additions.....	11
Chapter 2 Transaction and Terminal Profiles	12
2.1 Random Number Generation.....	12
2.2 Input/Output Screen Layout.....	12
2.3 Priced Terminal Feature Verification	12
2.4 Transaction Statistics	12
2.5 Presentation Manager or Intelligent Terminal.....	13
2.6 Queuing Mechanism	13
Chapter 3 Transaction and System Properties	14
3.1 Transaction System Properties (ACID Tests).....	14
3.2 Atomicity Tests.....	14
3.2.1 COMMIT Transaction	14
3.2.2 ROLLBACK Transaction.....	14
3.3 Consistency Tests	14
3.4 Isolation Tests.....	15
3.5 Durability Tests	15
3.5.1 Loss of Data / Loss of Log.....	15
3.5.2 Loss of System / Memory	16
Chapter 4 Scaling and Database Population.....	17
4.1 Database Layout	17
4.2 Initial Cardinality of Tables	21
4.3 60 Day Space	21
4.3.1 Transaction Log Space Requirements.....	21
4.4 Type of Database Used	22
4.5 Database Mapping.....	22
Chapter 5 Performance Metrics and Response Time	23
5.1 Throughput.....	23
5.2 Response Times.....	23
5.3 Keying and Think Times.....	23
5.4 Response Time Frequency.....	24
5.4.1 New Order Response Time.....	24
5.4.2 Payment Response Time Distribution.....	25

5.4.3	Order Status Response Time.....	26
5.4.4	Delivery Response Time Distribution	27
5.4.5	Stock Level Response Time.....	28
5.4.6	Response Time Versus Throughput	29
5.4.7	New Order Think Time Distribution.....	30
5.4.8	Throughput Versus Time Distribution	31
5.5	Steady State Determination.....	31
5.6	Work Performed During Steady State.....	31
5.6.1	Checkpoint.....	31
5.6.2	Checkpoint Conditions	32
5.6.3	Checkpoint Implementation.....	32
5.7	Measurement Period Duration.....	32
5.8	Regulation of Transaction Mix	32
5.9	Transaction Mix.....	33
5.10	Transaction Statistics.....	33
5.11	Checkpoint Count and Location	33
Chapter 6	SUT, Driver and Communications Definition.....	34
6.1	RTE Description.....	34
6.2	Emulated Components	34
6.3	Functional Diagram.....	34
6.4	Networks.....	34
6.5	Operator Intervention	34
Chapter 7	Pricing.....	35
7.1	System Pricing.....	35
7.2	General Availability, Throughput and Price Performance	35
7.3	Country Specific Pricing	35
7.4	Usage Pricing.....	35
Chapter 8	Audit	36
8.1	Auditor's Information.....	36
Appendix A	Source Code	39
Appendix B	Database Load.....	109
B.1	Database Options.....	140
B.2	Table definitions	140
B.3	Stored Procedures	142
Appendix C	Tunable Parameters.....	195
C.1	Microsoft SQL Server 8.0 Configuration Parameters	245
C.2	Client System Configuration Parameters	246
C.3	RTE Input Parameters.....	263
Appendix D	60 Day Space Requirements	289
Appendix E	3 rd Party Pricing.....	290

Preface

Document Structure

This is the full disclosure report for a benchmark test of the HP Integrity Superdome using Microsoft SQL Server 2000 Enterprise Edition 64-bit. It meets the requirements of the TPC Benchmark ® C Standard Specification, Revision 5.1 dated December 2002. TPC Benchmark® C was developed by the Transaction Processing Performance Council (TPC). It is the intent of this group to develop a suite of benchmarks to measure the performance of computer systems executing a wide range of applications. Hewlett-Packard Company and Microsoft, Inc. are active participants in the TPC.

TPC Benchmark® C Overview

TPC Benchmark ® C is an **On Line Transaction Processing (OLTP)** workload. It is a mixture of read-only and update intensive transactions that mimic the activities found in complex OLTP application environments. It does so by exercising a breadth of system components associated with such environments, which are characterized by:

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

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

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

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

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

System Overview

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

The processor architecture of the HP Integrity Superdome was designed for the Intel Itanium 2 6M processor. The HP Integrity Superdome used in this test was powered by 64 1.5GHz Intel Itanium 2 6M processors, each with 6MB of 3rd level cache.

This configuration used 512 GB of HP SDRAM. This was achieved by using 512 1 GB DIMMs.

The operating system, all executables and libraries, the master database, and swap space were contained in two 18GB hard disk, attached to a PCI SCSI controller.

The database log drives storage was located on HP VA7410 diskarray. The Disk Array held 60 72GB 10krpm Fibrechannel hard drives. The VA7410 disk array was connected to the Superdome using 2 Emulex 9802 Fibrechannel HBA's. The disks were configured as RAID 0, the disk array controller cache was enabled.

The TPC-C database storage consisted of 1792 HP 18GB 15krpm hard drives. 64 HP StorageWorks MSA 1000 disk arrays were used to connect the disks. Each MSA1000 holds 14 disks, additionally one HP Storageworks enclosure model 4354R holding 14 disks was connected to each MSA1000 using two Ultra-3 SCSI channels.

The disks were configured as one RAID 0 volume over all 28 disks (per MSA1000).

The MSA1000 disk arrays were connected to the Superdome using 16 QLogic QL2340 HBA's and 6 QLogic SANbox2/16 Fibrechannel switches. Each HBA was connected to 4 MSA1000 disk arrays, zoning was used to separate the Fibrechannel Fabrics on the SANbox switch.

Each of the 32 clients is a HP Proliant DL360 G3 with two Intel Xeon at 2.8 GHz, 1024 MB RAM and one 36.1GB SCSI hard disk, running Microsoft Windows 2000 Server.

The server and web-clients were networked together using QLogic VIA/SAN. The network bandwidth between the web-clients and the database server was 2GBps. 64 remote terminal emulators (RTEs) emulated 635,920 users executing the standard TPC-C workload. Each web-client had two embedded Gigabit LAN adapters, one of which was used to connect to the RTEs running in 100MBit mode.

General Items

Test Sponsor

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

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

Application Code and Definition Statements

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

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

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

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

Parameter Settings

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

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

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

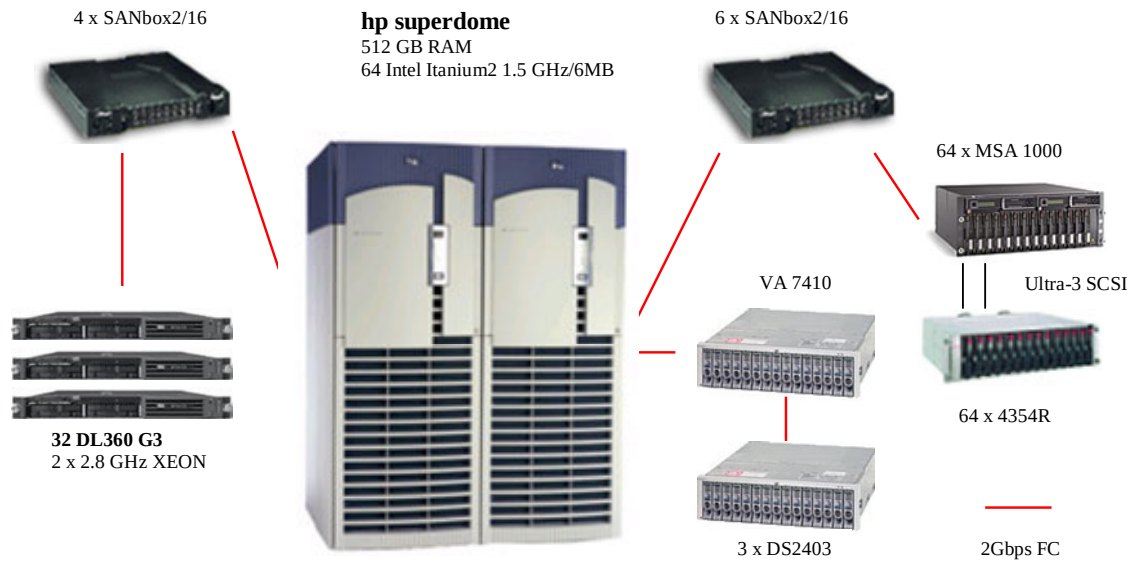
Appendix D contains the 60 day space calculations.

Configuration Diagrams

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

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

Figure 1. Measured and Priced Configuration



The measured configuration used a mix of Ultra-320 and discontinued Ultra-3 SCSI disks, Ultra320 SCSI disk were priced.

Chapter 1 Logical Database Design

1.1 Table Definitions

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

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

1.2 Physical Organization of the Database

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

The measured database configuration used a total of 1854 disks, which included 1792 18GB Hot Swap disk drives for data, 60 72GB drives for log, and two 18GB drive for the operating system.

Each of the 64 database disk arrays was Configured as 1 RAID 0 volume over 28 18GB drives. Each volume held 2 partitions, one for the CS filegroup where the Customer and Stock tables were stored and one partition for MISC filegroup where all other tables were stored.

1.3 Insert and Delete Operations

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

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

1.4 Partitioning

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

Partitioning was not used on any table.

1.5 Replication, Duplication or Additions

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

No replications, duplications or additional attributes were used.

Chapter 2 Transaction and Terminal Profiles

2.1 Random Number Generation

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

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

2.2

2.3 Input/Output Screen Layout

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

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

2.4 Priced Terminal Feature Verification

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

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

2.5 Transaction Statistics

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

Table 1 shows the transaction statistics.

Table 1. Transaction Statistics

Type	Item	Value
New Order	Home warehouse items	99.00%
	Remote warehouse items	1.00%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse	85.00%
	Remote warehouse	15.00%
	Non primary key access	60.01%
Order Status	Non primary key access	60.05%
Delivery	Skipped transactions	0
Transaction Mix	New Order	44.96%
	Payment	43.01%
	Delivery	4.01%
	Stock Level	4.01%
	Order Status	4.01%

2.6 Presentation Manager or Intelligent Terminal

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

Comment 1: *The intent of this clause is to describe any special manipulations performed by a local terminal or workstation to off-load work from the SUT. This includes, but is not limited to: screen presentations, message bundling, and local storage of TPC-C rows.*

Comment 2: *This disclosure also requires that all data manipulation functions performed by the local terminal to provide navigational aids for transaction(s) must also be described. Within this disclosure, the purpose of such additional function(s) must be explained.*

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

2.7 Queuing Mechanism

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

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

The source code is listed in Appendix A.

Chapter 3 Transaction and System Properties

3.1 Transaction System Properties (ACID Tests)

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

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

All of the specified ACID tests were successfully performed on the HP Integrity Superdome. A fully scaled database was used for the System Loss tests, the loss of Log and Data were performed on a 7200 warehouse database.

3.2 Atomicity Tests

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

3.2.1 COMMIT Transaction

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

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

3.2.2 ROLLBACK Transaction

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

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

3.3 Consistency Tests

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

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

3.4 Isolation Tests

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

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

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

3.5 Durability Tests

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

Three types of failures were tested to ensure the durability of the database: Loss of Data, Loss of Log, and Loss of System/Memory. A fully scaled database was used for the System Loss tests, the loss of Log and Data were performed on a 7200 warehouse database.

3.5.1 Loss of Log, loss of Data

The standard driving mechanism was used to generate the transaction load of 72000 users for the test. To demonstrate recovery from a permanent failure of durable media containing TPC-C tables, the following steps were executed:

1. The database was backed up using SQLServer backup facilities.
2. A sum of D_NEXT_O_ID was taken.
3. 72000 users were logged in to the database and ran transactions.
4. After 5 minutes, one of the (mirrored) log disk was removed from the system, processing transactions continued.
5. After another 5 minutes, one data disk drive was removed. The system stopped processing transactions.
6. The RTE was shutdown and a success file was created.
7. The Database was recovered by backing up the log, restoring the database and applying the log backup.
8. A new count of D_NEXT_O_ID was taken
9. This number was compared with the number of new orders reported by the RTE
10. Samples were taken of the RTE log and verified against the database.

3.5.2 Loss of System / Memory

This was demonstrated on the full database with 63,592 warehouses in a single test. The standard driving mechanism was used to generate the transaction load of 635,920 users for this test. To demonstrate recovery the following steps were followed:

1. The full database was used.
2. A sum of D_NEXT_O_ID was taken.
3. 635,920 users were logged in to the database and ran transactions.
4. After 5 minutes, power to the system was reset using the built-in Management Processor
5. The RTE continued running and completed transactions enroute from the clients were recorded. Error messages began appearing on the RTE screen.
6. The RTE was stopped.
7. The server machine rebooted.
8. Microsoft SQL Server was restarted and performed an automatic recovery.
9. A new count of D_NEXT_O_ID was taken.
10. This number was compared with the number of new orders reported by the RTE
11. Samples were taken of the RTE log and verified against the database.

Chapter 4 Scaling and Database Population

4.1 Database Layout

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

The measured (tested) and priced systems used 64 HP StorageWorks MSA 1000 Disk Arrays. These Disk arrays were connected using 16 QLogic FC HBA's. These cards were plugged into 2 PCI-X slots of each of the 8 IO chassis. The system drive was attached to a Ultra3 LVD SCSI controller. The VA7410 disk array that holds the database log was connected to 2 Emulex 9802 FC HBA's which were plugged into 2 PCI-X slots in the UIO chassis connected to cell 0.

The measured database configuration used a total of 1854 disks, which included 1792 18GB Hot Swap disk drives for data, 60 72GB drives for log, and two 18GB drives for the operating system. Each of the 64 database disk arrays was Configured as 1 RAID 0 volume over 28 18GB drives. Each volume held 2 partitions, One for the CS filegroup where the Customer and Stock tables were stored and one partition for MISC filegroup where all other tables were stored.

Table 2 shows the complete data distribution.

Table 2: Data Distribution

Qlogic 2340 HBA #1					WINDOWS.NET DISK ADMIN	
MSA #1					DISK 1	
SCSI ID	Channels				Partitions (RAID 0):107GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\CS1 Raw 65250MB	G:\MNT\MISC1 Raw 36250MB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
7	18GB	18GB	18GB	18GB		
MSA #2					WINDOWS.NET DISK ADMIN	
					DISK 2	
SCSI ID	Channels				Partitions (RAID 0):107GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\CS2 Raw 65250MB	G:\MNT\MISC2 Raw 36250MB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
7	18GB	18GB	18GB	18GB		
MSA #3					WINDOWS.NET DISK ADMIN	
					DISK 3	
SCSI ID	Channels				Partitions (RAID 0):107GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\CS3 Raw 65250MB	G:\MNT\MISC3 Raw 36250MB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
7	18GB	18GB	18GB	18GB		
MSA #4					WINDOWS.NET DISK ADMIN	
					DISK 4	
SCSI ID	Channels				Partitions (RAID 0):107GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\CS4 Raw 65250MB	G:\MNT\MISC4 Raw 36250MB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
7	18GB	18GB	18GB	18GB		

Qlogic 2340 HBA #2					WINDOWS.NET DISK ADMIN	
MSA #5					DISK 5	
SCSI ID	Channels				Partitions (RAID 0):107GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\CS5 Raw 65250MB	G:\MNT\MISC5 Raw 36250MB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
7	18GB	18GB	18GB	18GB		
MSA #6					WINDOWS.NET DISK ADMIN	
					DISK 6	
SCSI ID	Channels				Partitions (RAID 0):107GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\CS6 Raw 65250MB	G:\MNT\MISC6 Raw 36250MB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
7	18GB	18GB	18GB	18GB		
MSA #7					WINDOWS.NET DISK ADMIN	
					DISK 7	
SCSI ID	Channels				Partitions (RAID 0):107GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\CS7 Raw 65250MB	G:\MNT\MISC7 Raw 36250MB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
7	18GB	18GB	18GB	18GB		
MSA #8					WINDOWS.NET DISK ADMIN	
					DISK 8	
SCSI ID	Channels				Partitions (RAID 0):107GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\CS8 Raw 65250MB	G:\MNT\MISC8 Raw 36250MB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
7	18GB	18GB	18GB	18GB		

....

Qlogic 2340 HBA #16					WINDOWS.NET DISK ADMIN	
MSA #61					DISK 61	
SCSI ID	Channels				Partitions (RAID 0):107GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\CS61 Raw 65250MB	G:\MNT\MISC61 Raw 36250MB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
7	18GB	18GB	18GB	18GB		
MSA #62					WINDOWS.NET DISK ADMIN	
					DISK 62	
SCSI ID	Channels				Partitions (RAID 0):107GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\CS62 Raw 65250MB	G:\MNT\MISC62 Raw 36250MB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
7	18GB	18GB	18GB	18GB		
MSA #63					WINDOWS.NET DISK ADMIN	
					DISK 1	
SCSI ID	Channels				Partitions (RAID 0):107GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\CS63 Raw 65250MB	G:\MNT\MISC63 Raw 36250MB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
7	18GB	18GB	18GB	18GB		
MSA #64					WINDOWS.NET DISK ADMIN	
					DISK 1	
SCSI ID	Channels				Partitions (RAID 0):107GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\CS64 Raw 65250MB	G:\MNT\MISC64 Raw 36250MB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
7	18GB	18GB	18GB	18GB		

4.2 Initial Cardinality of Tables

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

Table 2 shows the cardinality of the various tables.

Table 3: Table Cardinality

Table	Occurrences
Warehouse	72,000
District	720,000
Customer	2,160,000,000
History	2,160,000,000
Orders	2,160,000,000
New Orders	648,000,000
Order Line	18,000,014,496
Item	100,000
Stock	7,200,000,000

No rows were deleted for the benchmark runs.

4.3 60 Day Space

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

4.3.1 Transaction Log Space Requirements

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

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

The result of the above steps yielded a requirement of 3,939GB to sustain the log for 8 hours. Space available for the transaction log was 4,260 GB indicating that enough storage was configured to hold 8 hours of growth.

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

4.4 Type of Database Used

A statement must be provided that describes:

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

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

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

4.5 Database Mapping

The mapping of database partitions and replications must be described.

The database was divided into 2 file groups misc and cs. Misc consists of 64 files at 36150 MBeach and cs consists of 64 files at 62150 MB each as shown in the createdb.sql. The log was configured with 800,000 MB.

Chapter 5 Performance Metrics and Response Time

5.1 Throughput

Measured tpmC® must be reported.

Measured TpmC®: 786,646.62
Price per TpmC®: \$6.49

5.2 Response Times

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

Table 3 shows the response times for all transaction types.

Table 4: Transaction Response Times

Response Times	Average	90th %-ile	Maximum
New-Order	0.69s	1.33s	56.51s
Payment	0.65s	1.30s	28.21s
Order-Status	0.67s	1.33s	28.29s
Delivery (interactive portion)	0.10s	0.11s	6.51s
Delivery (deferred portion)	0.10s	0.15s	3.13s
Stock-Level	0.87s	1.56s	10.19s
Menu	0.10s	0.11s	13.62s

5.3 Keying and Think Times

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

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

Table 5: Transaction Key Times

Keying Times	Minimum	Average	Maximum
New Order	18s	18.02s	18.03s
Payment	3s	3.02s	3.03s
Order Status	2s	2.02s	2.03s
Interactive Delivery	2s	2.02s	2.03s
Stock Level	2s	2.02s	2.03s

Table 6: Transaction Think Times

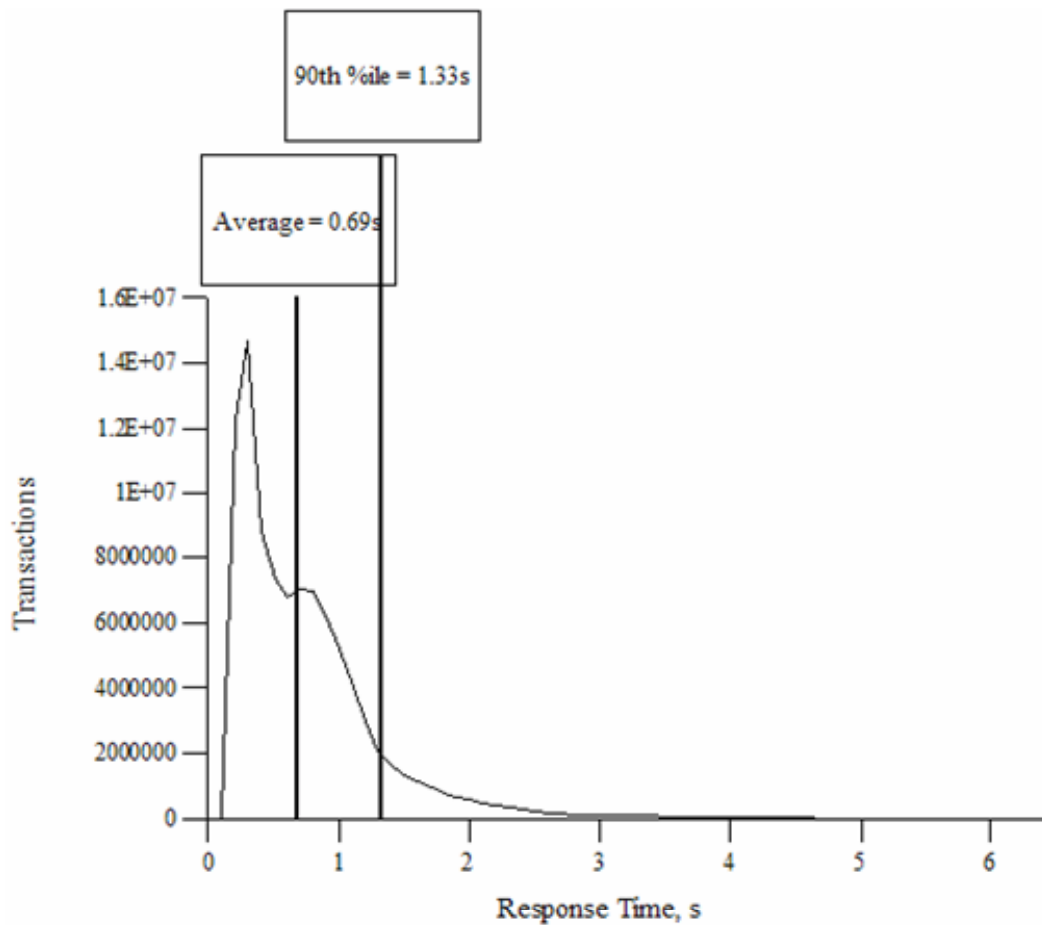
Think Times	Minimum	Average	Maximum
New Order	0s	12.05s	120.51s
Payment	0s	12.04s	120.51s
Order Status	0s	10.05s	100.51s
Interactive Delivery	0s	5.06s	50.51s
Stock Level	0s	5.05s	50.51s

5.4 Response Time Frequency

Response Time frequency distribution curves (see Clause 5.6.1) must be reported for each transaction type. The performance curve for response times versus throughput (see Clause 5.6.2) must be reported for the New-Order transaction. Think Time frequency distribution curves (see Clause 5.6.3) must be reported for each transaction type. Keying Time frequency distribution curves (see Clause 5.6.4) must be reported for each transaction type. A graph of throughput versus elapsed time (see Clause 5.6.5) must be reported for the New-Order transaction.

5.4.1 New Order Response Time

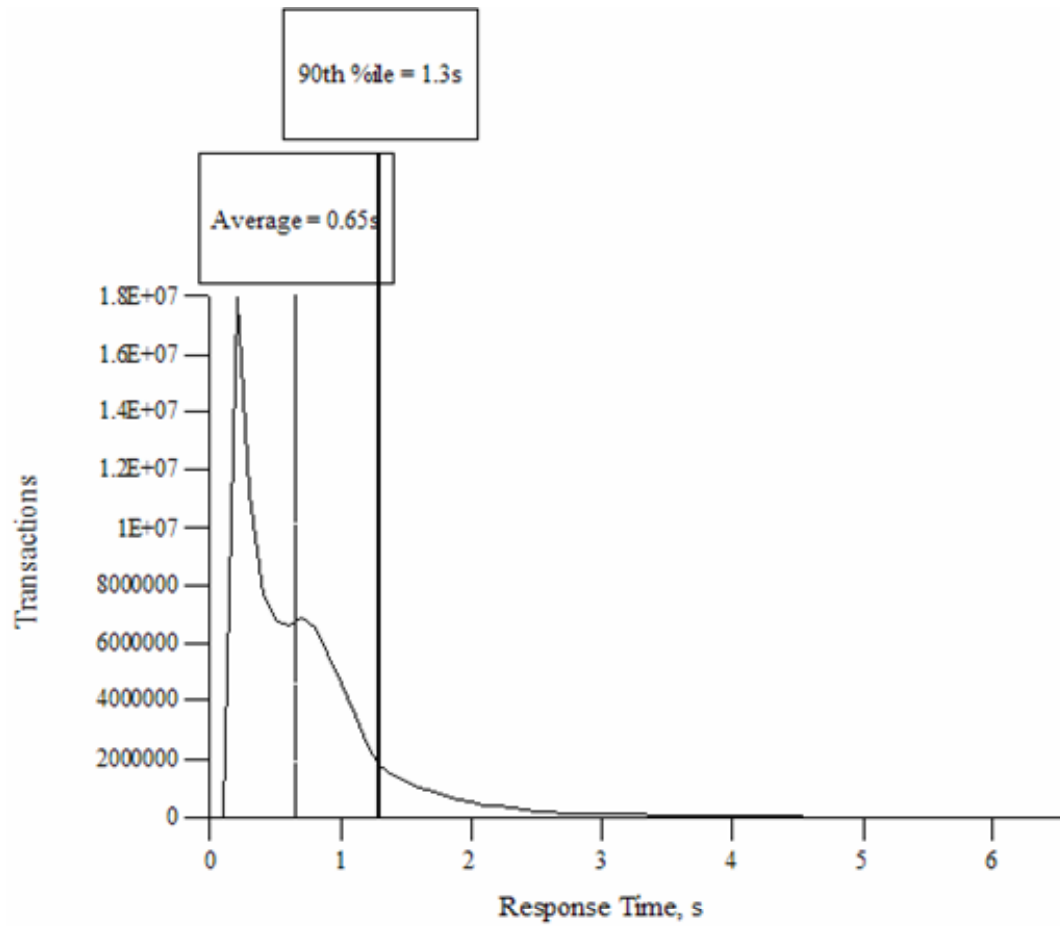
Figure 3: New Order Response Time Distribution



Response time frequency distribution for New Order transaction

5.4.2 Payment Response Time Distribution

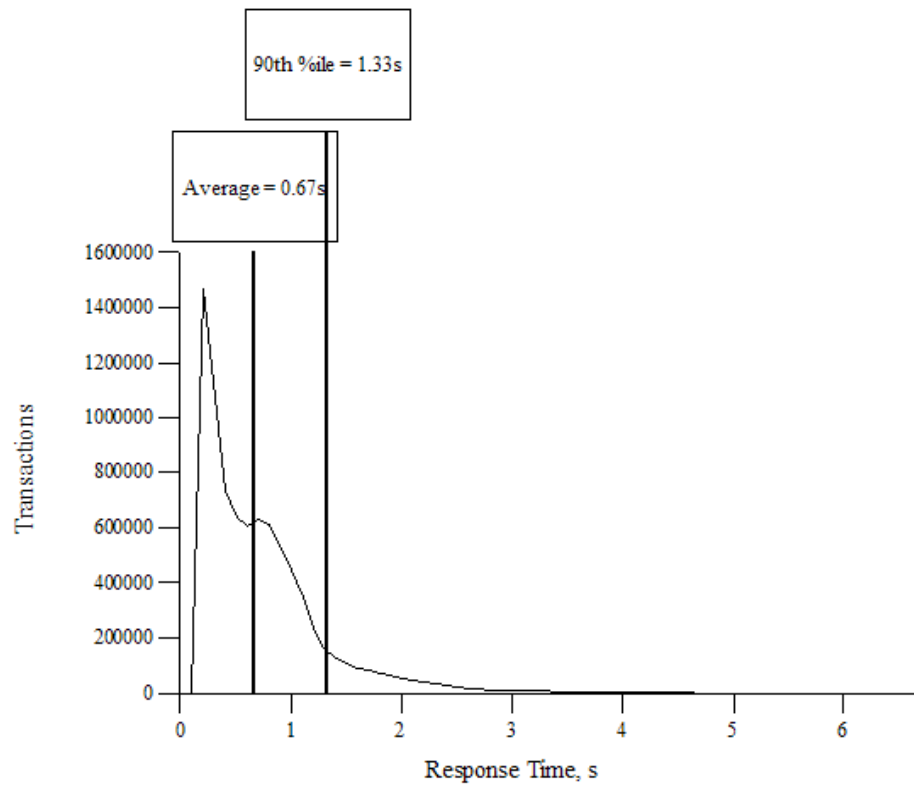
Figure 4: Payment Response Time Distribution



Response time frequency distribution for Payment transaction

5.4.3 Order Status Response Time

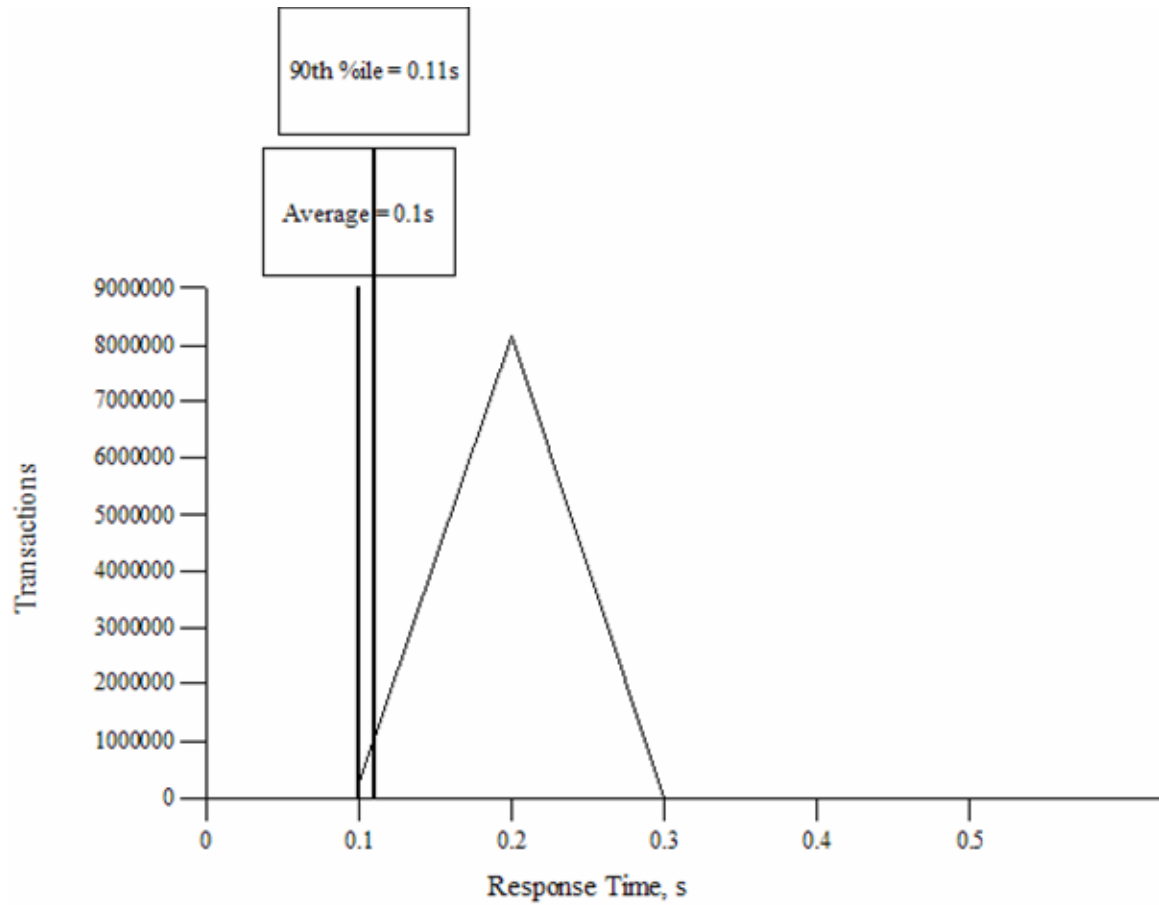
Figure 5: Order Status Response Time Distribution



Response time frequency distribution for Order Status transaction

5.4.4 Delivery Response Time Distribution

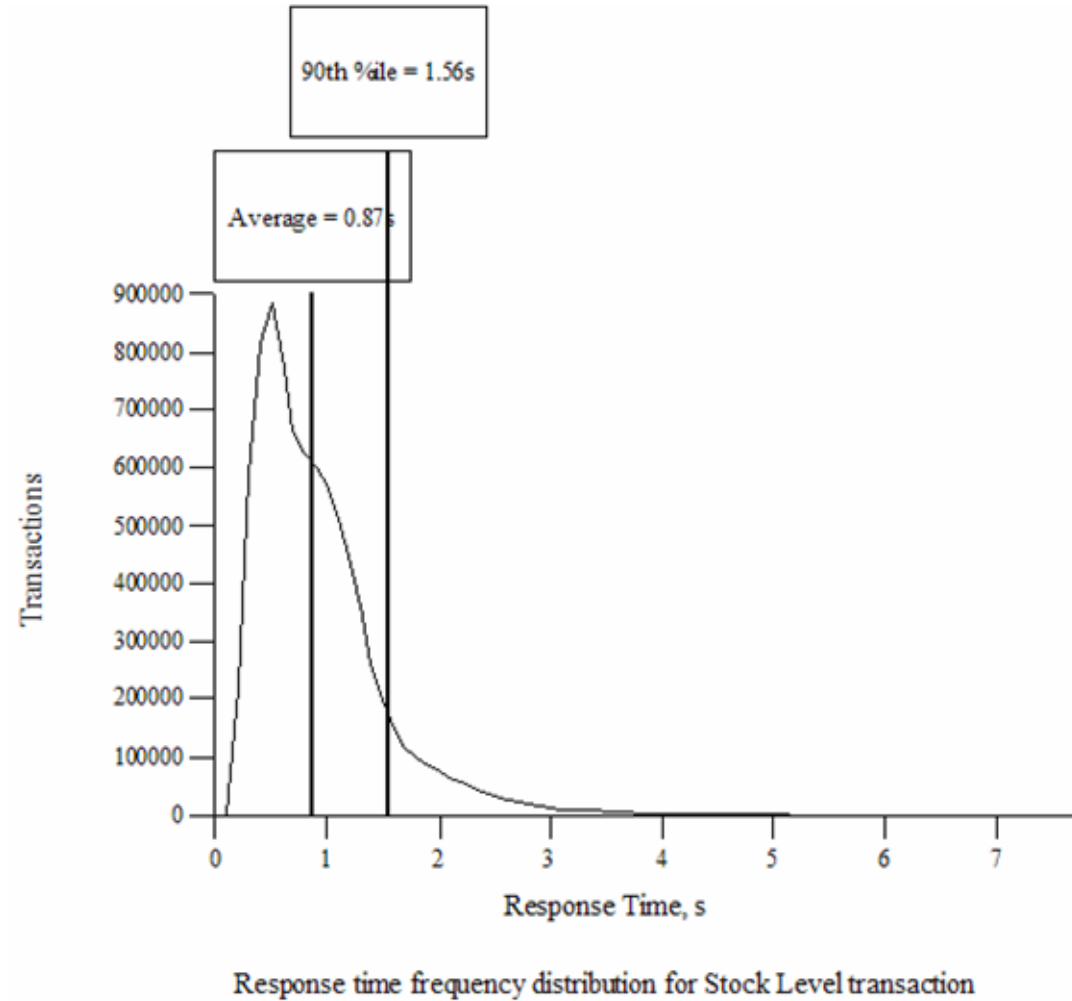
Figure 6: Delivery Response Time Distribution



Response time frequency distribution for Delivery transaction

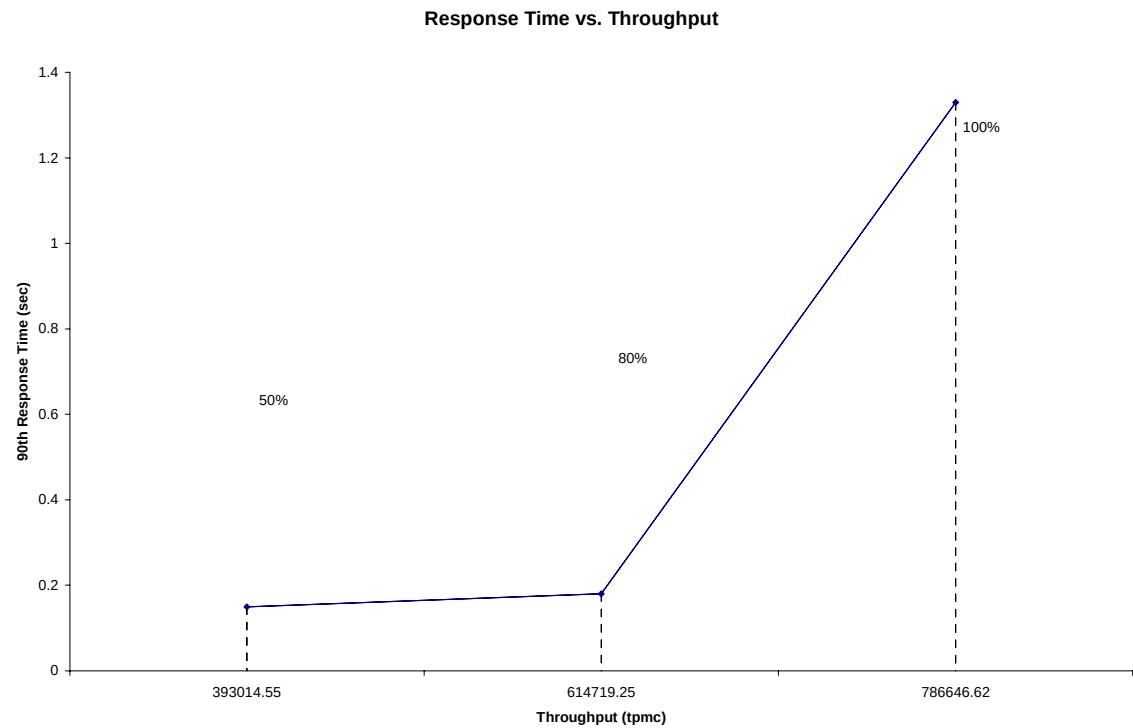
5.4.5 Stock Level Response Time

Figure 7: Stock Level Response Time Distribution



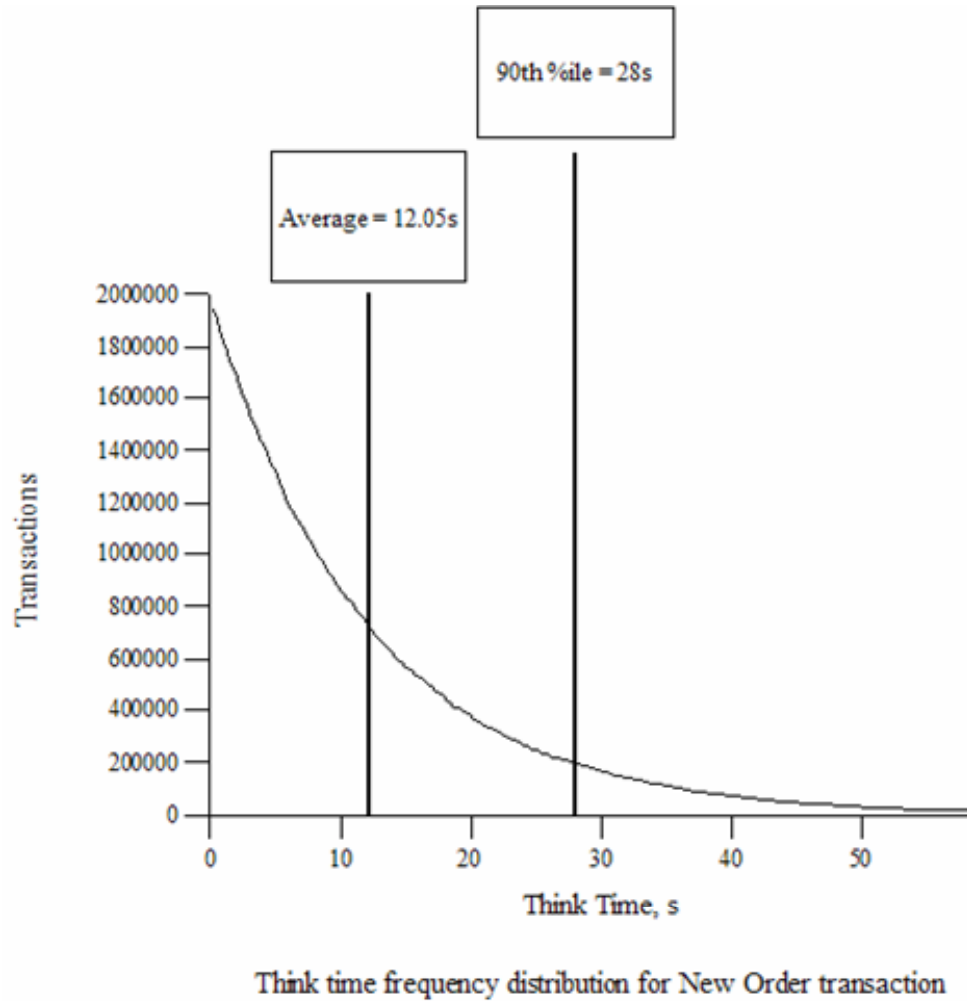
5.4.6 Response Time Versus Throughput

Figure 8: New Order Response Time Distribution



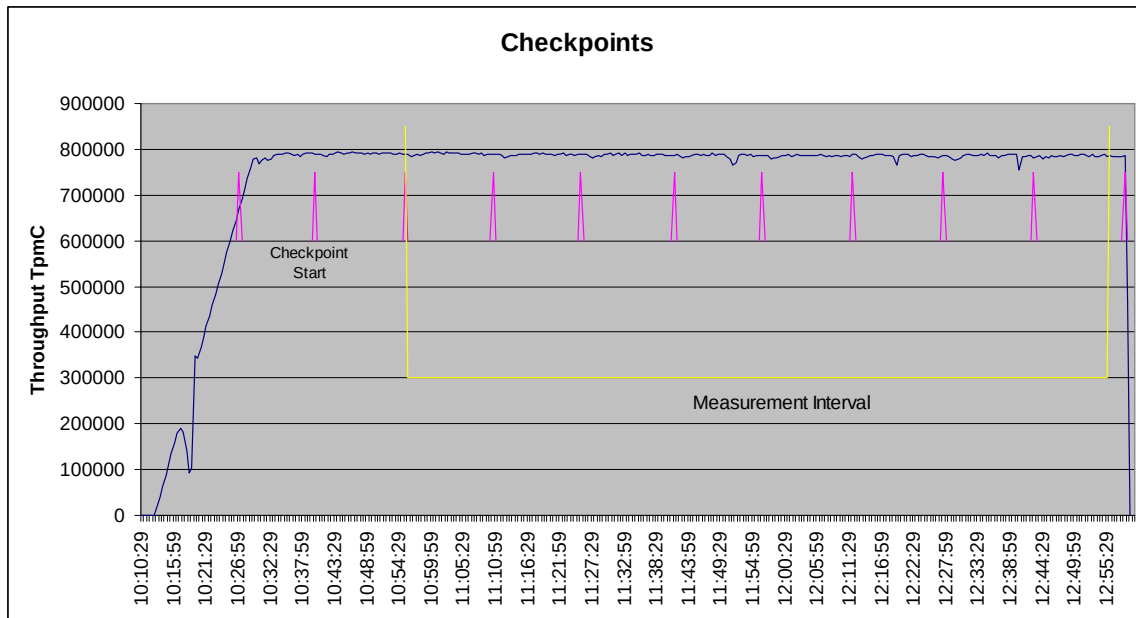
5.4.7 New Order Think Time Distribution

Figure 9: New Order Think Time Distribution



5.4.8 Throughput Versus Time Distribution

Figure 10: New Order Throughput versus Time



5.5 Steady State Determination

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

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

5.6 Work Performed During Steady State

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

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

5.6.1 Checkpoint

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

5.6.2 Checkpoint Conditions

There are two types of checkpoints:

1. Checkpoints that are executed automatically by SQL Server.
2. Checkpoints that are forced by database owners with the CHECKPOINT statement.

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

5.6.3 Checkpoint Implementation

A Windows command script was issued to start manual checkpoints back to back., the average duration of a checkpoint was 15 minutes. The traceflag T828 was used. This causes SQL server to issue writes at maximum queue depth for the checkpoint. The checkpoints were affinitized to CPU 63 using connection affinity. A dedicated QL2350 VI card was used, the connection affinity for this card was set to CPU 63.

By setting the TRACE FLAG #3502, SQL Server logged the checkpoint beginning and ending time in the ERRORLOG file.

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

5.7 Measurement Period Duration

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

The measurement interval was 120 minutes.

5.8 Regulation of Transaction Mix

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

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

5.9 Transaction Mix

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

Table 7: Transaction Mix

Type	Percentage
New Order	44.96%
Payment	43.01%
Delivery	4.01%
Stock Level	4.01%
Order Status	4.01%

5.10 Transaction Statistics

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

Table 1 contains the required items.

5.11 Checkpoint Count and Location

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

The measurement interval is 120 minutes. There are 8 checkpoints within the measurement interval and 4 checkpoint before the measurement interval.

Table 8: Measurement Interval and Checkpoints

Event	From	To
Measured Interval	10:55:18	12:55:18
Checkpoint	10:19:03	10:26:03
Checkpoint	10:26:36	10:40:03
Checkpoint	10:40:03	10:55:16
Checkpoint	10:55:16	11:10:36
Checkpoint	11:10:36	11:25:48
Checkpoint	11:25:48	11:41:11
Checkpoint	11:41:11	11:56:34
Checkpoint	11:56:34	12:11:52
Checkpoint	12:11:52	12:27:25
Checkpoint	12:27:25	12:42:54
Checkpoint	12:42:54	12:58:22

Chapter 6 SUT, Driver and Communications Definition

6.1 RTE Description

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

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

6.2 Emulated Components

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

No components were emulated.

6.3 Functional Diagram

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

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

6.4 Networks

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

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

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

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

6.5 Operator Intervention

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

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

Chapter 7 Pricing

7.1 System Pricing

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

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

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

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

7.2 General Availability, Throughput and Price Performance

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

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

Table 9: Throughput, Price Performance and Availability

Maximum qualified throughput:	786646.62 tpmC
Price per tpmC:	\$6.49
Availability:	October 23, 2003

7.3 Country Specific Pricing

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

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

7.4 Usage Pricing

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

The component pricing based on usage is shown below:

- 64 Microsoft SQL Server 2000 Enterprise Edition 64-bit per-processor licenses.
- Microsoft Windows Server 2003, Datacenter edition (64-bit)
- 32 Microsoft Windows 2000 Server licenses.
- 1 Microsoft Visual C++ 32bit Edition.
- 3 year support for hardware components

Chapter 8 Audit

8.1 Auditor's Information

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

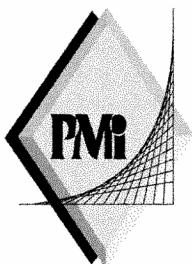
The test methodology and results of this TPC Benchmark C were audited by:

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

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

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

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



PERFORMANCE METRICS INC.
TPC Certified Auditors

August 25, 2003

Mr. Gunter Zink
ISS Redmond Performance
Hewlett-Packard Company
14475 NE 24th St.
Bellevue, WA 98007

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

Platform: Hewlett-Packard SuperDome Enterprise Server
Database Manager: Microsoft SQL Server 2000 Enterprise Edition 64 bit
Operating System: Microsoft Windows Server 2003, Datacenter Edition 64 bit
Transaction Monitor: Microsoft COM+

Servers: Hewlett-Packard SuperDome Enterprise Server with:				
CPU's	Memory	Disks (total)	90% Response	TpmC
64 Itanium 2 @ 1.5 Ghz	Main: 512 GB	60 @ 72 GB 1792 @ 18 GB 2 @ 18 GB (os)	1.33	786,646.62
32 Clients: ProLiant DL360 G3 each with:				
2 Pentium Xeon @ 2.8 Ghz	Main: 1 GB	1 @ 36 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 files were properly sized and populated for 72,000 warehouses, of which 63,592 were active during the measured interval.
- Inactive warehouses were verified to be unchanged during the performance run.
- The ACID properties were successfully demonstrated.

PERFORMANCE METRICS INC.
TPC Certified Auditors

- The durability data loss and log loss tests were demonstrated on a subset of the SUT configured with a database properly populated for 7,200 warehouses.
- Input data was generated according to the specified percentages.
- Eight hours of mirrored log space was present on the measured system.
- Eight hours of growth space for the dynamic tables was present on the measured system.
- The data for the 60 day space calculation was verified. Disk substitution was used to meet the storage requirements. See auditor's notes below.
- The Measured cycle times were confirmed to have the correct response time delays.
- There were 635,920 user contexts present on the system.
- Each emulated users started with a different random number seed.
- The NURand constants used for C_last load and run were 123 and 233 respectively.
- The steady state portion of the test was 120 minutes.
- More than one checkpoint was taken after steady state and before the measured interval.
- Seven checkpoints were contained completely inside the measured interval, and additional checkpoints continued after the measured interval closed.
- Checkpoint interval was verified to be approximately 15 minutes.
- The system pricing was checked for major components and maintenance.
- Third party quotes were verified for compliance.

Auditor Notes:

The measured configuration had 1 full array populated with 28 of the priced disks. The rest of the configuration was running older, obsolete disks. The priced disks had the same technical specifications and capacities as the obsolete disks. This substitution is compliant with the substitution rules.

None.

Sincerely,



Lorna Livingtree
Auditor

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

Page 2

Appendix A Source Code

Isapi_dll/src/tpcc.def

LIBRARY TPCC.DLL

EXPORTS

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

Isapi_dll/src/tpcc.h

```
/*      FILE:          TPCC.H
 *
 *      Ver. 4.20.000
 *
 *      Copyright Microsoft,
 *      1999
 *      All Rights Reserved
 *
 *      Version 4.10.000
 *      audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *      PURPOSE:   Header file for ISAPI TPCC.DLL, defines structures
 *      and functions used in the isapi tpcc.dll.
 *
 */

//VERSION RESOURCE DEFINES
#define _APS_NEXT_RESOURCE_VALUE
101
#define _APS_NEXT_COMMAND_VALUE
40001
#define _APS_NEXT_CONTROL_VALUE
1000
#define _APS_NEXT_SYMED_VALUE
101

#define TP_MAX_RETRIES

50
```

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

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

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

    CTPCC_BASE *pTxn;
} CLIENTDATA, *PCLIENTDATA;
```

```
//This structure is used to define the operational interface for terminal id
typedef struct _TERM
{
    int iNumEntries;
    //total allocated
    terminal array entries
    int iFreeList;
    //next
    available terminal array element or -1 if none
    int iMasterSyncId;
    //synchronization id
    CLIENTDATA *pClientData;
    //pointer to allocated client data
} TERM;

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

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

```

ERR_NEWORDER_SUPPW_INVALID,
ERR_NO_SERVER_SPECIFIED,
ERR_ORDERSTATUS_CID_AND_CLT,
ERR_ORDERSTATUS_CID_INVALID,
ERR_ORDERSTATUS_CLT_RANGE,
ERR_ORDERSTATUS_DID_INVALID,

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

};

class CWEBCLNT_ERR : public CBaseErr
{
public:

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

    CWEBCLNT_ERR(WEBERROR Err, char
*szTextDetail, DWORD dwSystemErr)
    {
        m_Error = Err;
        m_szTextDetail = new
char[strlen(szTextDetail)+1];

```

```

);
        strcpy( m_szTextDetail, szTextDetail
        m_SystemErr = dwSystemErr;
        m_szErrorText = NULL;
    };

    ~CWEBCLNT_ERR()
    {
        if (m_szTextDetail != NULL)
            delete []
            m_szTextDetail;

        if (m_szErrorText != NULL)
            delete []
            m_szErrorText;
    };

    WEBERROR m_Error;
    char m_szTextDetail;

    // char m_szErrorText;
    DWORD m_SystemErr;

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

};

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

//function prototypes

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call,
LPVOID lpReserved);
void WriteMessageToEventLog(LPTSTR lpszMsg);
void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd,
int *pFormId, int *pTermId, int *pSyncId);
void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId);
void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId);
void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int iError, int
iErrorType, char *szMsg, int iTermId);
void GetKeyValue(char **pQueryString, char *pKey, char *pValue, int iMax,
WEBERROR err);

```

```

int GetIntKeyValue(char **pQueryString, char *pKey, WEBERROR NoKeyErr,
WEBERROR NoValErr);
void TermDeleteAll(void);
int TermAdd(void);
void TermDelete(int id);
void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int
iErrorNum, int iTermId, int iSyncId, char *szErrorText, char *szBuffer );
void MakeMainMenuForm(int iTermId, int iSyncId, char *szForm);
void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA
*pStockLevelData, BOOL bInput, char *szForm);
void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA *pNewOrderData,
BOOL bInput, char *szForm);
void MakePaymentForm(int iTermId, PAYMENT_DATA *pPaymentData, BOOL
bInput, char *szForm);
void MakeOrderStatusForm(int iTermId, ORDER_STATUS_DATA
*pOrderStatusData, BOOL bInput, char *szForm);
void MakeDeliveryForm(int iTermId, DELIVERY_DATA *pDeliveryData, BOOL
bInput, char *szForm);
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK *pECB,
int iTermId, char *szBuffer);
void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId,
char *szBuffer);
void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA
*pNewOrderData);
void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA
*pPaymentData);
void GetOrderStatusData(LPSTR lpszQueryString, ORDER_STATUS_DATA
*pOrderStatusData);
BOOL PostDeliveryInfo(long w_id, short o_carrier_id);
BOOL IsNumeric(char *ptr);
BOOL IsDecimal(char *ptr);
void DeliveryWorkerThread(void *ptr);

```

Isapi_dll/src/tpcc.rc

```

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

#define APSTUDIO_READONLY_SYMBOLS
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 2 resource.

```

```
//
#include "afxres.h"

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

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

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

#ifdef _MAC
////////////////////////////////////
//
// Version
//

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

```

```
END

#ifdef ! _MAC

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

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

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

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

#endif // APSTUDIO_INVOKED

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

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

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

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE

```

```
END
BEGIN
    LEFTMARGIN, 7
    RIGHTMARGIN, 179
    TOPMARGIN, 7
    BOTTOMMARGIN, 88
END
END
#endif // APSTUDIO_INVOKED

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

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

```

Isapi_dll/src/tpcc.cpp

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

#include <windows.h>
#include <process.h>
#include <tchar.h>
#include <stdio.h>
#include <stdarg.h>

```

```

#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sys/timeb.h>
#include <io.h>
#include <assert.h>

#include <sqltypes.h>

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

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

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

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

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

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

#define LEN_ERR_STRING      256

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

char
szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

```

```

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

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

static      CRITICAL_SECTION      TermCriticalSection;

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

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

// For deferred Delivery txns:

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

HANDLE
hWorkerSemaphore = INVALID_HANDLE_VALUE;

HANDLE
hDoneEvent =
INVALID_HANDLE_VALUE;

HANDLE
*pDeliHandles = NULL;

// configuration settings from registry
TPCCREGISTRYDATA      Reg;

DWORD
dwNumDeliveryThreads = 4;
CRITICAL_SECTION      DelBuffCriticalSection;
//critical section for delivery transactions cache
DELIVERY_TRANSACTION *pDelBuff = NULL;

DWORD
dwDelBuffSize = 100;
// size of circular buffer for delivery txns

DWORD
dwDelBuffFreeCount;
// number of buffers free

```

```

DWORD
dwDelBuffBusyIndex = 0;           // index
position of entry waiting to be delivered
dwDelBuffFreeIndex = 0;           // index
position of unused entry

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

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

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

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

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

                GetComputerName(szMyComputerName, &dwSize);

                szMyComputerName[dwSize] = 0;
            }
        }
    }
}

```

<pre> DisableThreadLibraryCalls((HMODULE)hModule); InitializeCriticalSection(&TermCriticalSection); if (ReadTPCCRegistrySettings(&Reg)) throw new CWEBCLNT_ERR(ERR_MISSING_REGISTRY_ENTRIES); dwDelBuffSize = min(Reg.dwMaxPendingDeliveries, 10000); // min with 10000 as a sanity constraint dwNumDeliveryThreads = min(Reg.dwNumberOfDeliveryThreads, 100); // min with 100 as a sanity constraint TermInit(); // load DLL for txn monitor TUXEDO) { strcpy(szDllName, Reg.szPath); strcat(szDllName, "tpcc_tuxedo.dll"); hLibInstanceTm = LoadLibrary(szDllName); if (hLibInstanceTm == NULL) throw new CWEBCLNT_ERR(ERR_LOADDLL_FAILED, szDllName, GetLastError()); // get function pointer to wrapper for class constructor pCTPCC_TUXEDO_new = (TYPE_CTPCC_TUXEDO*) GetProcAddress(hLibInstanceTm,"CTPCC_TUXEDO_new"); if (pCTPCC_TUXEDO_new == NULL) throw new CWEBCLNT_ERR(ERR_GETPROCADDR_FAILED, szDllName, GetLastError()); } else if (Reg.eTxnMon == ENCINA) { </pre>	<pre> strcpy(szDllName, Reg.szPath); strcat(szDllName, "tpcc_encina.dll"); hLibInstanceTm = LoadLibrary(szDllName); if (hLibInstanceTm == NULL) throw new CWEBCLNT_ERR(ERR_LOADDLL_FAILED, szDllName, GetLastError()); // get function pointer to wrapper for class constructor pCTPCC_ENCINA_new = (TYPE_CTPCC_ENCINA*) GetProcAddress(hLibInstanceTm,"CTPCC_ENCINA_new"); pCTPCC_ENCINA_post_init = (TYPE_CTPCC_ENCINA*) GetProcAddress(hLibInstanceTm,"CTPCC_ENCINA_post_init"); if (pCTPCC_ENCINA_new == NULL) throw new CWEBCLNT_ERR(ERR_GETPROCADDR_FAILED, szDllName, GetLastError()); } else if (Reg.eTxnMon == COM) { strcpy(szDllName, Reg.szPath); strcat(szDllName, "tpcc_com.dll"); hLibInstanceTm = LoadLibrary(szDllName); if (hLibInstanceTm == NULL) throw new CWEBCLNT_ERR(ERR_LOADDLL_FAILED, szDllName, GetLastError()); // get function pointer to wrapper for class constructor pCTPCC_COM_new = (TYPE_CTPCC_COM*) GetProcAddress(hLibInstanceTm,"CTPCC_COM_new"); if (pCTPCC_COM_new == NULL) throw new CWEBCLNT_ERR(ERR_GETPROCADDR_FAILED, szDllName, GetLastError()); } // load DLL for database connection </pre>	<pre> if ((Reg.eTxnMon == None) (dwNumDeliveryThreads > 0)) { if (Reg.eDB_Protocol == DBLIB) { strcpy(szDllName, Reg.szPath); strcat(szDllName, "tpcc_dblib.dll"); hLibInstanceDb = LoadLibrary(szDllName); if (hLibInstanceDb == NULL) throw new CWEBCLNT_ERR(ERR_LOADDLL_FAILED, szDllName, GetLastError()); // get function pointer to wrapper for class constructor pCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*) GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_new"); if (pCTPCC_DBLIB_new == NULL) throw new CWEBCLNT_ERR(ERR_GETPROCADDR_FAILED, szDllName, GetLastError()); } else if (Reg.eDB_Protocol == ODBC) { strcpy(szDllName, Reg.szPath); strcat(szDllName, "tpcc_odbc.dll"); hLibInstanceDb = LoadLibrary(szDllName); if (hLibInstanceDb == NULL) throw new CWEBCLNT_ERR(ERR_LOADDLL_FAILED, szDllName, GetLastError()); // get function pointer to wrapper for class constructor pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*) GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new"); if (pCTPCC_ODBC_new == NULL) throw new CWEBCLNT_ERR(ERR_GETPROCADDR_FAILED, szDllName, GetLastError()); </pre>
---	---	---

```

    }
}

if
(dwNumDeliveryThreads)
{
    // for
deferred delivery txns:

    hDoneEvent = CreateEvent( NULL, TRUE /* manual reset */,
FALSE /* initially not signalled */, NULL );

    InitializeCriticalSection(&DelBuffCriticalSection);

    hWorkerSemaphore = CreateSemaphore( NULL, 0,
dwDelBuffSize, NULL );

    dwDelBuffFreeCount = dwDelBuffSize;

    InitJulianTime(NULL);

    // create
unique log file name based on delilog-yyymmdd-hhmm.log

    SYSTEMTIME Time;

    GetLocalTime( &Time );

    wsprintf(
szLogFile, "%sdelivery-%2.2d%2.2d%2.2d-%2.2d%2.2d.log",
Reg.szPath, Time.wYear % 100, Time.wMonth,
Time.wDay, Time.wHour, Time.wMinute );

    txnDelilog
= new CTxnLog(szLogFile, TXN_LOG_WRITE);

    //write
event into txn log for START

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

    //
allocate structures for delivery buffers and thread mgmt

    pDeliHandles = new HANDLE[dwNumDeliveryThreads];
pDelBuff
= new DELIVERY_TRANSACTION[dwDelBuffSize];

    // launch
DeliveryWorkerThread to perform actual delivery txns

    for(i=0;
i<dwNumDeliveryThreads; i++)

```

```

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

    if (pDeliHandles[i] == INVALID_HANDLE_VALUE)

        throw new CWEBCLNT_ERR(
ERR_DELIVERY_THREAD_FAILED );

    }

    break;

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

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

            // This will do a clean shutdown of the delivery log file

            CTxnLog *txnDelilogLocal = txnDelilog;

            txnDelilog= NULL;

            delete txnDelilogLocal;

        }

        delete []
pDeliHandles;

        delete []
pDelBuff;

        CloseHandle( hWorkerSemaphore );

        CloseHandle( hDoneEvent );

        DeleteCriticalSection(&DelBuffCriticalSection);
    }

    DeleteCriticalSection(&TermCriticalSection);
}

```

```

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

    hLibInstanceTm =
NULL;

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

    hLibInstanceDb =
NULL;

    Sleep(500);
    break;

default:
    /* nothing */;
}

}
catch (CBaseErr *e)
{
    WriteMessageToEventLog( e->ErrorText() );
    delete e;
    TerminateExtension(0);
    return FALSE;
}
catch (...)
{
    WriteMessageToEventLog(TEXT("Unhandled
exception. DLL could not load."));
    TerminateExtension(0);
    return FALSE;
}

return TRUE;
}

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

BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO *pVer)
{

```

```

        pVer->dwExtensionVersion =
        MAKELONG(HSE_VERSION_MINOR, HSE_VERSION_MAJOR);
        lstrcpy(pVer->lpExtensionDesc, "TPC-C Server.",
        HSE_MAX_EXT_DLL_NAME_LEN);

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

        return TRUE;
    }

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

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

    TermDeleteAll();
    return TRUE;
}

/* FUNCTION: HttpExtensionProc
*
* PURPOSE: This function is the main entry point for the TPCC DLL. The
internet service
*
* calls this function passing in the http
string.
*
* ARGUMENTS: EXTENSION_CONTROL_BLOCK *pECB
*
* structure pointer to passed in internet
*
*
* service information.
*

```

```

* RETURNS: DWORD HSE_STATUS_SUCCESS
connection can be
*
* dropped if error HSE_STATUS_SUCCESS_AND_KEEP_CONN keep
connect valid comment sent
*
* COMMENTS: None
*
*/

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

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

#ifdef ICECAP
    StartCAP();
#endif

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

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

                WriteMessageToEventLog( szTmp );

                throw new
CWEBCLNT_ERR( ERR_INVALID_TERMID );
            }

            //must have a valid syncid here
            if (iSyncId !=
Term.pClientData[TermId].iSyncId)

```

```

throw new
CWEBCLNT_ERR( ERR_INVALID_SYNC_CONNECTION );
//set use time

Term.pClientData[TermId].iTickCount = GetTickCount();
}

switch(iCmd)
{
case 0:
    WelcomeForm(pECB, szBuffer);
    break;
case 1:
    switch( FormId )
    {
        case WELCOME_FORM:
            break;
MAIN_MENU_FORM:
            break;
NEW_ORDER_FORM:
            ProcessNewOrderForm(pECB, TermId, szBuffer);
            break;
            case PAYMENT_FORM:
                ProcessPaymentForm(pECB, TermId, szBuffer);
                break;
            case DELIVERY_FORM:
                ProcessDeliveryForm(pECB, TermId, szBuffer);
                break;
ORDER_STATUS_FORM:
            ProcessOrderStatusForm(pECB, TermId, szBuffer);
            break;
            case
STOCK_LEVEL_FORM:
                ProcessStockLevelForm(pECB, TermId, szBuffer);
                break;
    }
    break;
case 2:
    // new-order selected from menu;
    display new-order input form
    MakeNewOrderForm(TermId, NULL,
INPUT_FORM, szBuffer);
    break;
case 3:

```

```

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

```

```

ErrorForm( pECB, e->ErrorType(), e->ErrorNum(),
TermId, iSyncId, e->ErrorText(), szBuffer );
}
catch (...)
{
ErrorForm( pECB, ERR_TYPE_WEBDLL, 0, TermId,
iSyncId, "Error: Unhandled exception in Web Client.", szBuffer );
}
#endif
#ifdef ICECAP
StopCAP();
#endif
lpbSize = strlen(szBuffer);
wsprintf(szHeader1,
"Content-Type: text/html\r\n"
"Content-Length: %d\r\n"
"Connection: Keep-Alive\r\n\r\n",
lpbSize);
strcat( szHeader1, szBuffer );
(*pECB->ServerSupportFunction)(pECB->ConnID,
HSE_REQ_SEND_RESPONSE_HEADER, szHeader, (LPDWORD) &dwSize,
(LPDWORD)szHeader1);
//finish up and keep connection
pECB->dwHttpStatusCode = 200;
return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}
void WriteMessageToEventLog(LPTSTR lpszMsg)
{
TCHAR szMsg[256];
HANDLE hEventSource;
LPTSTR lpszStrings[2];
// Use event logging to log the error.
//
hEventSource = RegisterEventSource(NULL, TEXT("TPCC.DLL"));
_stprintf(szMsg, TEXT("Error in TPCC.DLL: "));
lpszStrings[0] = szMsg;
lpszStrings[1] = lpszMsg;
if (hEventSource != NULL)
{
ReportEvent(hEventSource, // handle of event source
EVENTLOG_ERROR_TYPE, // event type
0, // event category
0, // event ID
NULL, // current user's SID

```

```

0, // no bytes of raw data
(LPCTSTR *)lpszStrings, // array of error strings
NULL); // no raw data
}
}
(VOID) DeregisterEventSource(hEventSource);
}
}
/* FUNCTION: DeliveryWorkerThread
*
* PURPOSE: This function processes deferred delivery txns. There are
typically several threads running this routine. The
number of threads is determined by an entry
read from the registry. The thread
waits for work by waiting on semaphore.
* When a delivery txn is posted, the
semaphore is released. After processing
the delivery txn, information is
logged to record the txn status and execution
time.
*/
/*static*/ void DeliveryWorkerThread(void *ptr)
{
CTPCC_BASE *pTxn = NULL;
DELIVERY_TRANSACTION delivery;
PDELIVERY_DATA
pDeliveryData;
TXN_RECORD_TPCC_DELIV_DEF txnDeliRec;
DWORD
index;
HANDLE
handles[2];
SYSTEMTIME trans_end; //delivery
transaction finished time
SYSTEMTIME trans_start; //delivery transaction
start time
assert(txnDeliRec != NULL);
try
{
if (Reg.eDB_Protocol == ODBC)
pTxn = pCTPCC_ODBC_new(
Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, szMyComputerName,
Reg.szDbName );
else if (Reg.eDB_Protocol == DBLIB)

```



```

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

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

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

```

```

        // make a local copy of
        current entry from delivery buffer and increment buffer index
        EnterCriticalSection(&DelBuffCriticalSection);
        delivery =
        *(pDelBuff+dwDelBuffBusyIndex);

        dwDelBuffFreeCount++;

        dwDelBuffBusyIndex++;
        if (dwDelBuffBusyIndex
        == dwDelBuffSize) // wrap-around if at end of buffer
            dwDelBuffBusyIndex = 0;

        LeaveCriticalSection(&DelBuffCriticalSection);

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

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

        GetLocalTime(
        &trans_start );
        pTxn->Delivery();
        GetLocalTime(
        &trans_end );
        //log txn
        txnDeliRec.TxnStatus =
        ERR_SUCCESS;
        for (int i=0; i<10; i++)

            txnDeliRec.o_id[i] = pDeliveryData->o_id[i];
            txnDeliRec.DeltaT4 =
            (int)(Get64BitTime(&trans_end) - txnDeliRec.TxnStartT0);

            txnDeliRec.DeltaTxnExec = (int)(Get64BitTime(&trans_end) -
            Get64BitTime(&trans_start));

            if (txnDeliLog != NULL)
                txnDeliLog->WriteToLog(&txnDeliRec);
            }
            catch (CBaseErr *e)

```

```

        {
            char szTmp[1024];
            wsprintf( szTmp, "Error in Delivery
Txn thread. %s", e->ErrorText() );
            WriteMessageToEventLog( szTmp );

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

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

        ErrorExit:
        delete pTxn;
        _endthread();
    }

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

    BOOL PostDeliveryInfo(long w_id, short o_carrier_id)
    {
        BOOL bError;

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

```

```

>queue);

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

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

return bError;
}

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

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

    //allowable client command strings i.e. CMD=command
    static char *szCmds[] =

```

```

        "Process", "..NewOrder..", "..Payment..",
        "..Delivery..", "..Order-Status..", "ShowMenu", "Menu", "Clear", "Stats", ""
    };

    *pCmd = 0; // default is the login

screen
    *pTermId = 0;

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

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

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

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

/* FUNCTION: void WelcomeForm
*
*
*/

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

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

```

```

        "<B><BIG>Microsoft TPC-C Web Client (ver 4.20)</BIG></B>"
    <BR> <BR>"
    "<font face=\"Courier New\"><PRE>"

    "Compiled: " __DATE__ ", " __TIME__ " <BR>"

    "Source: " __FILE__ " (" __TIMESTAMP__ ") <BR>"

    "</PRE></font>"

    "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"

    "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"  
VALUE=\"0\">"

    "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"

    "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"1\">"

    "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"0\">"

    "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"0\">"

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

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

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

```

```

    }
    strcat( szBuffer, "</PRE></font>");

    if (Reg.eTxnMon == None)
        // connection options may be specified when not
using a txn monitor
        sprintf( szTmp, "Please enter your
database options for this connection:<BR>"

        "<font face=\"Courier New\" color=\"blue\"><PRE>"

        "DB Server  = <INPUT NAME=\"db_server\" SIZE=20
VALUE=\"%s\"><BR>"

        "DB User ID  = <INPUT NAME=\"db_user\" SIZE=20
VALUE=\"%s\"><BR>"

        "DB Password = <INPUT NAME=\"db_passwd\" SIZE=20
VALUE=\"%s\"><BR>"

        "DB Name     = <INPUT NAME=\"db_name\" SIZE=20
VALUE=\"%s\"><BR>"

        "</PRE></font>"

    Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
    else
        // if using a txn monitor, connection options are
determined from registry; can't
        // set per user. show options fyi
        sprintf( szTmp, "Database options
which will be used by the transaction monitor:<BR>"

        "<font face=\"Courier New\" color=\"blue\"><PRE>"

        "DB Server      = <B>%s</B><BR>"

        "DB User ID      = <B>%s</B><BR>"

        "DB Password     = <B>%s</B><BR>"

        "DB Name         = <B>%s</B><BR>"

        "</PRE></font>"

    Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
    strcat( szBuffer, szTmp);

    sprintf( szTmp, "Please enter your Warehouse and
District for this session:<BR>"

    "<font
face=\"Courier New\" color=\"blue\"><PRE>" );

```

```

    strcat( szBuffer, szTmp); "Warehouse ID = <INPUT
NAME=\"w_id\" SIZE=6><BR>"

    "District ID = <INPUT NAME=\"d_id\" SIZE=2><BR>"

    "</PRE></font><HR>"

    "<INPUT TYPE=\"submit\" NAME=\"CMD\""
VALUE=\"Submit\">"
}

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

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

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

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

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

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

```

```

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

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

    iNewTerm = TermAdd();

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

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

    upward

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

/* FUNCTION: StatsCmd

```

```

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

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

    EnterCriticalSection(&TermCriticalSection);

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

    LeaveCriticalSection(&TermCriticalSection);

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

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

```

```

{
    ERR_DELIVERY_MISSING_OCD_KEY,
    "Delivery missing Carrier ID key \"OCD*\"."
},
{
    ERR_DELIVERY_THREAD_FAILED,
    "Could not start
delivery worker thread."
},
{
    ERR_GETPROCADDR_FAILED,
    "Could
not map proc in DLL. GetProcAddr error. DLL="
},
{
    ERR_HTML_ILL_FORMED,
    "Required key field is
missing from HTML string."
},
{
    ERR_INVALID_SYNC_CONNECTION,
    "Invalid Terminal Sync ID."
},
{
    ERR_INVALID_TERMID,
    "Invalid
Terminal ID."
},
{
    ERR_LOADDLL_FAILED,
    "Load of
DLL failed. DLL="
},
{
    ERR_MAX_CONNECTIONS_EXCEEDED,
    "No connections available. Max Connections is probably too
low." },
{
    ERR_MISSING_REGISTRY_ENTRIES,
    "Required registry entries are
missing. Rerun INSTALL to correct."
},
{
    ERR_NEWORDER_CUSTOMER_INVALID,
    "New Order customer id invalid data type, range = 1 to 3000."
},
{
    ERR_NEWORDER_CUSTOMER_KEY,
    "New Order missing
Customer key \"CID*\"."
},
{
    ERR_NEWORDER_DISTRICT_INVALID,
    "New Order District ID Invalid range 1 - 10."
},
}

```

```

{
    ERR_NEWORDER_FORM_MISSING_DID,
    "New Order missing District key \"DID*\"."
},
{
    ERR_NEWORDER_ITEMID_INVALID,
    "New Order Item Id is wrong data
type, must be numeric."
},
{
    ERR_NEWORDER_ITEMID_RANGE,
    "New Order Item Id is
out of range. Range = 1 to 999999."
},
{
    ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
    "New Order Item_Id field entered without a corresponding
Supp_W."
},
{
    ERR_NEWORDER_MISSING_IID_KEY,
    "New Order missing Item Id key \"IID*\"."
},
{
    ERR_NEWORDER_MISSING_QTY_KEY,
    "New Order Missing Qty key \"Qty##*\"."
},
{
    ERR_NEWORDER_MISSING_SUPPW_KEY,
    "New Order missing Supp_W key \"SP##*\"."
},
{
    ERR_NEWORDER_NOITEMS_ENTERED,
    "New Order No order lines entered."
},
{
    ERR_NEWORDER_QTY_INVALID,
    "New Order Qty invalid
must be numeric range 1 - 99."
},
{
    ERR_NEWORDER_QTY_RANGE,
    "New
Order Qty is out of range. Range = 1 to 99."
},
{
    ERR_NEWORDER_QTY_WITHOUT_SUPPW,
    "New Order Qty field entered without a
corresponding Supp_W."
},
{
    ERR_NEWORDER_SUPPW_INVALID,
    "New Order Supp_W
invalid data type must be numeric."
},
}

```

```

        {
            ERR_NO_SERVER_SPECIFIED,
            "No Server name
specified."
        },
        {
            ERR_ORDERSTATUS_CID_AND_CLT,
            "Order Status Only Customer ID or
Last Name may be entered, not both."
        },
        {
            ERR_ORDERSTATUS_CID_INVALID,
            "Order Status Customer ID invalid,
range must be numeric 1 - 3000."
        },
        {
            ERR_ORDERSTATUS_CLT_RANGE,
            "Order Status
Customer last name longer than 16 characters."
        },
        {
            ERR_ORDERSTATUS_DID_INVALID,
            "Order Status District invalid, value
must be numeric 1 - 10."
        },
        {
            ERR_ORDERSTATUS_MISSING_CID_CLT,
            "Order Status Either Customer ID or Last Name must be
entered."
        },
        {
            ERR_ORDERSTATUS_MISSING_CID_KEY,
            "Order Status missing Customer key \"CID*\"."
        },
        {
            ERR_ORDERSTATUS_MISSING_CLT_KEY,
            "Order Status missing Customer Last Name key \"CLT*\"."
        },
        {
            ERR_ORDERSTATUS_MISSING_DID_KEY,
            "Order Status missing District key \"DID*\"."
        },
        {
            ERR_PAYMENT_CDI_INVALID,
            "Payment Customer
district invalid must be numeric."
        },
        {
            ERR_PAYMENT_CID_AND_CLT,
            "Payment Only
Customer ID or Last Name may be entered, not both."
        },
        {
            ERR_PAYMENT_CUSTOMER_INVALID,
            "Payment Customer data type invalid, must be numeric."
        },
        {
            ERR_PAYMENT_CWI_INVALID,
            "Payment Customer
Warehouse invalid, must be numeric."
        },
        {
            ERR_PAYMENT_DISTRICT_INVALID,
            "Payment District ID is invalid, must
be 1 - 10."
        }
    },
    {
        ERR_PAYMENT_HAM_INVALID,
        "Payment Amount
invalid data type must be numeric."
    },
    {
        ERR_PAYMENT_HAM_RANGE,
        "Payment
Amount out of range, 0 - 9999.99."
    },
    {
        ERR_PAYMENT_LAST_NAME_TO_LONG,
        "Payment Customer last name longer than 16 characters."
    },
    {
        ERR_PAYMENT_MISSING_CDI_KEY,
        "Payment missing Customer district
key \"CDI*\"."
    },
    {
        ERR_PAYMENT_MISSING_CID_CLT,
        "Payment Either Customer ID or Last
Name must be entered."
    },
    {
        ERR_PAYMENT_MISSING_CID_KEY,
        "Payment missing Customer Key
\"CID*\"."
    },
    {
        ERR_PAYMENT_MISSING_CLT_KEY,
        "Payment missing Customer Last
Name key \"CLT*\"."
    },
    {
        ERR_PAYMENT_MISSING_CWI_KEY,
        "Payment missing Customer
Warehouse key \"CWI*\"."
    },
    {
        ERR_PAYMENT_MISSING_DID_KEY,
        "Payment missing District Key
\"DID*\"."
    },
    {
        ERR_PAYMENT_MISSING_HAM_KEY,
        "Payment missing Amount key
\"HAM*\"."
    },
    {
        ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
        "Stock
Level; missing Threshold key \"TT*\"."
    },
    {
        ERR_STOCKLEVEL_THRESHOLD_INVALID,
        "Stock Level; Threshold value must be in the range = 1 - 99."
    },
    {
        ERR_STOCKLEVEL_THRESHOLD_RANGE,
        "Stock Level Threshold out of range, range must
be 1 - 99."
    },
    {
        ERR_VERSION_MISMATCH,
        "Invalid
version field. RTE and Web Client are probably out of sync."
    }
}

```

```

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

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

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

/* FUNCTION: GetKeyValue
*
* PURPOSE: This function parses a http formatted string for specific key
values.
*
* ARGUMENTS:      char      http string from client browser
                  *pQueryString
                  char      key value
                  *pKey
to look for

```

```

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

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

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

/* FUNCTION: GetKeyValue
*
* PURPOSE: This function parses a http formatted string for specific key
values.
*
* ARGUMENTS:      char      http string from client browser
                  *pQueryString
                  char      key value
                  *pKey
to look for

```

```

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

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

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

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

    *pQueryString = ptr;
    return;

ErrorExit:

```

```

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

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

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

```

```

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

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

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

    *pQueryString = ptr;
    return atoi(ptr0);

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

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

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

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

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

```

```

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

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

LeaveCriticalSection(&TermCriticalSection);
}

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

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

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

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

    LeaveCriticalSection(&TermCriticalSection);
}

/* FUNCTION: TermAdd
*

```

```

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

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

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

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

        Term.pClientData[iNewTerm].iTickCount = GetTickCount();
    }
}

```

```

Term.pClientData[iNewTerm].iSyncId =
Term.iMasterSyncId;
Term.pClientData[iNewTerm].pTxn = NULL;

LeaveCriticalSection(&TermCriticalSection);
return iNewTerm;
}

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

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

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

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

        LeaveCriticalSection(&TermCriticalSection);
    }
}

/* FUNCTION: MakeErrorForm
*/

void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int
iErrorNum, int iTermId, int iSyncId, char *szErrorText, char *szBuffer )
{
    wsprintf(szBuffer,
        "<HTML><HEAD><TITLE>TPC-C
Error</TITLE></HEAD><BODY>"
        "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
        "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"
    );
}

```

```

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

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

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

    c = sprintf(szForm,
        "<HTML><HEAD><TITLE>TPC-C Stock Level</TITLE></HEAD><FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
        "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"
        "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
        "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
        "<PRE><font face=\"Courier\">"
        "Stock-Level<BR>"
        "Warehouse: %6.6d District: %2.2d<BR>"
        "
        STOCK_LEVEL_FORM, iTermId,
        Term.pClientData[iTermId].iSyncId,
        Term.pClientData[iTermId].w_id,
        Term.pClientData[iTermId].d_id);

    if ( bInput )
    {
        strcpy(szForm+c,
            "Stock Level Threshold: <INPUT NAME=\"TT*\" SIZE=2><BR> <BR>"
            "low stock: </font><BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"

```

54


```

}

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

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

    c = sprintf(szForm,
                "<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD><BODY>"
                "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
                "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">"
                "<INPUT TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"0\">"
                "<INPUT TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\" NAME=\"TERMID\"
VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\" NAME=\"SYNCD\"
VALUE=\"%d\">"
                "<PRE><font face=\"Courier\">
Payment<BR>"
                "Date: "
                , pPaymentData->h_date.year,
                pPaymentData->h_date.month,
                pPaymentData->h_date.day,
                pPaymentData->h_date.hour,
                pPaymentData->h_date.minute,
                pPaymentData->h_date.second);

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

    if ( bInput )
    {
        c += sprintf(szForm+c,
                    "<BR><BR>Warehouse: %6.6d"

```

56


```

*
* browser.
*
* ARGUMENTS: EXTENSION_CONTROL_BLOCK *pECB
* passed in structure pointer from inetsrv.
*
* int
*
* client browser terminal id
*
*/

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

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

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

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

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

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

```

```

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

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

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

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

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

    PDELIVERY_DATA pDelivery;

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

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

    if (dwNumDeliveryThreads)
    {
        //post delivery info

```

```

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

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

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

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

    PSTOCK_LEVEL_DATA pStockLevel;

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

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

    pStockLevel->threshold = GetIntKeyValue(&ptr, "TT*",
ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
ERR_STOCKLEVEL_THRESHOLD_INVALID);

```

```

        if ( pStockLevel->threshold >= 100 || pStockLevel->threshold
< 0 )
            throw new CWBCLNT_ERR(
ERR_STOCKLEVEL_THRESHOLD_RANGE );

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

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

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

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

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

```

```

      "Qty10*", "Qty11*", "Qty12*", "Qty13*",
      "Qty14*" };
    pNewOrderData->d_id = GetIntKeyValue(&ptr, "DID*",
ERR_NEWORDER_FORM_MISSING_DID,
ERR_NEWORDER_DISTRICT_INVALID);
    pNewOrderData->c_id = GetIntKeyValue(&ptr, "CID*",
ERR_NEWORDER_CUSTOMER_KEY,
ERR_NEWORDER_CUSTOMER_INVALID);

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

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

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

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

            GetKeyValue(&ptr, szQty[i], szTmp,
sizeof(szTmp), ERR_NEWORDER_MISSING_QTY_KEY);

```

```

            if ( szTmp[0] ) throw new
CWBCLNT_ERR( ERR_NEWORDER_QTY_WITHOUT_SUPPW );
        }
    }
    if ( items == 0 )
        throw new CWBCLNT_ERR(
ERR_NEWORDER_NOITEMS_ENTERED );

    pNewOrderData->o_ol_cnt = items;
}

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

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

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

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

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

```

```

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

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

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

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

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

```

```

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

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

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

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

```

```

*
*           TRUE                if string contains only numeric characters i.e. '0' -
'9/'
*
BOOL IsNumeric(char *ptr)
{
    if ( *ptr == 0 )
        return FALSE;

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

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

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

    if ( *ptr == 0 )
        return FALSE;

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

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

```

```

        bValid = TRUE;

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

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

```

Isapi_dll/src/resource.h

```

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

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

```

common/src/ReadRegistry.h

```

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

```

```

 * Change history:      4.20.000 - first version
 */

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

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

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

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );

```

ReadRegistry.cpp

```

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

/* FUNCTION: ReadTPCCRegistrySettings

```

```

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

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

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

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

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

```



```

        if ( !strcmp(szTmp, "YES") )
            pReg->bCOM_SinglePool = TRUE;
    }

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

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

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

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

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

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

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

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

    RegCloseKey(hKey);

```

```

    }
    return FALSE;

```

common\src\error.h

```

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

#pragma once

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

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

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

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

#define ERR_TYPE_LOGIC      -1

error in program; internal error

```

```

#define ERR_SUCCESS          0
#define ERR_BAD_ITEM_ID      1 //success (a non-error error)
                                //expected abort record in txnRecord
#define ERR_TYPE_DELIVERY_POST 2 //expected delivery
post failed
#define ERR_TYPE_WEBDLL      3
                                //tpcc web generated error
#define ERR_TYPE_SQL         4
                                //sql server generated error
#define ERR_TYPE_DBLIB       5
                                //dblib generated error
#define ERR_TYPE_ODBC        6
                                //odbc generated error
#define ERR_TYPE_SOCKET      7
                                //error on communication socket client rte only
#define ERR_TYPE_DEADLOCK    8 //dblib
and odbc only deadlock condition
#define ERR_TYPE_COM         9 //error
from COM call
#define ERR_TYPE_TUXEDO      10
                                //tuxedo error
#define ERR_TYPE_OS          11
                                //operating system error
#define ERR_TYPE_MEMORY      12
                                //memory allocation error
#define ERR_TYPE_TPCC_ODBC   13 //error
from tpcc odbc txn module
#define ERR_TYPE_TPCC_DBLIB  14 //error
from tpcc dblib txn module
#define ERR_TYPE_DELISRV     15 //delivery
server error
#define ERR_TYPE_TXNLOG      16
                                //txn log error
#define ERR_TYPE_BCONN       17
                                //Benchcraft connection class

```

```

#define ERR_TYPE_TPCC_CONN      18
//Benchcraft connection class
#define ERR_TYPE_ENCINA        19
//Encina error
#define ERR_TYPE_COMPONENT     20 //error
from COM component

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

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

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

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

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

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

```

```

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

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

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

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

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

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

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

```

```

        if (m_szMsg) j += sprintf(szTmp+j, "%s\n",
m_szMsg);

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

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

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

class CSocketErr : public CBaseErr
{
public:
    enum Action
    {
        eNone,
        eSend,
        eSocket,
        eConnect
    };

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

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

class CSystemErr : public CBaseErr
{
public:
    enum Action
    {
        eNone,
        eTransactNamedPipe,
        eWaitNamedPipe,
        eSetNamedPipeHandleState,
        eCreateFile,
        eCreateProcess,
        eCallNamedPipe,
        eCreateEvent,

```

```

        eCreateThread,
        eVirtualAlloc,
        eReadFile,
        eWriteFile,
        eMapViewOfFile,
        eCreateFileMapping,
        eInitializeSecurityDescriptor,
        eSetSecurityDescriptorDacl,
        eCreateNamedPipe,
        eConnectNamedPipe,
    };

    CSystemErr(Action eAction, LPCTSTR szLocation);

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

    int m_errId;
    Action m_eAction;

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

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

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

```

common\src\trans.h

```

/*      FILE:      TRANS.H
 *
 *      Microsoft TPC-C Kit
 *
 *      Ver. 4.42.000
 *      Copyright Microsoft,
 *      2002
 *      All Rights Reserved
 *
 *      Version 4.10.000
 *      audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *      PURPOSE:   Header file for TPC-C structure templates.

```

```

 * Change history:
 *
 *      4.42.000 - changed w_id fields from short to long
 *      to support >32K warehouses
 *
 *      4.20.000 - updated rev number to match kit
 */
#pragma once

// String length constants
#define SERVER_NAME_LEN      20
#define DATABASE_NAME_LEN   20
#define USER_NAME_LEN       20
#define PASSWORD_LEN        20
#define TABLE_NAME_LEN     20
#define I_DATA_LEN          50
#define I_NAME_LEN          24
#define BRAND_LEN           1
#define LAST_NAME_LEN       16
#define W_NAME_LEN          10
#define ADDRESS_LEN         20
#define STATE_LEN           2
#define ZIP_LEN              9
#define S_DIST_LEN          24
#define S_DATA_LEN          50
#define D_NAME_LEN          10
#define FIRST_NAME_LEN      16
#define MIDDLE_NAME_LEN     2
#define PHONE_LEN           16
#define DATETIME_LEN        30
#define CREDIT_LEN           2
#define C_DATA_LEN          250
#define H_DATA_LEN          24
#define DIST_INFO_LEN       24
#define MAX_OL_NEW_ORDER_ITEMS 15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN          25
#define OL_DIST_INFO_LEN    24

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

```

```

    unsigned short /* hour;
    unsigned short /* minute;
    unsigned short /* second;
    unsigned long /* fraction;
    } TIMESTAMP_STRUCT;
#endif

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

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

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

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

    // output params
    EXEC_STATUS exec_status_code;
    char c_last[LAST_NAME_LEN+1];

```

```

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

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

    // output params
    EXEC_STATUS   exec_status_code;
    TIMESTAMP_STRUCT h_date;
    char          w_street_1[ADDRESS_LEN+1];
    char          w_street_2[ADDRESS_LEN+1];
    char          w_city[ADDRESS_LEN+1];
    char          w_state[STATE_LEN+1];
    char          w_zip[ZIP_LEN+1];
    char          d_street_1[ADDRESS_LEN+1];
    char          d_street_2[ADDRESS_LEN+1];
    char          d_city[ADDRESS_LEN+1];
    char          d_state[STATE_LEN+1];
    char          d_zip[ZIP_LEN+1];
    char          c_first[FIRST_NAME_LEN+1];
    char          c_middle[MIDDLE_NAME_LEN + 1];
    char          c_street_1[ADDRESS_LEN+1];

```

```

char          c_street_2[ADDRESS_LEN+1];
char          c_city[ADDRESS_LEN+1];
char          c_state[STATE_LEN+1];
char          c_zip[ZIP_LEN+1];
char          c_phone[PHONE_LEN+1];
TIMESTAMP_STRUCT c_since;
char          c_credit[CREDIT_LEN+1];
double        c_credit_lim;
double        c_discount;
double        c_balance;
char          c_data[200+1];
} PAYMENT_DATA, *PPAYMENT_DATA;

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

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

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

typedef struct
{
    // input params
    long          w_id;
    short         o_carrier_id;

```

```

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

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

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

    // output params
    EXEC_STATUS   exec_status_code;
    long          low_stock;
} STOCK_LEVEL_DATA, *PSTOCK_LEVEL_DATA;common\src\txn_base.h
/* FILE: TXN_BASE.H
*
* Microsoft TPC-C Kit
Ver. 4.20.000
* Copyright Microsoft,
1999
* All Rights Reserved
*
* Version 4.10.000
audited by Richard Gimarc, Performance Metrics, 3/17/99
*
* PURPOSE: Header file for TPC-C txn class implementation.
*
* Change history:
* 4.20.000 - updated rev number to match kit
*/

#pragma once

// need to declare functions for import, unless define has already been
created
// by the DLL's .cpp module for export.
#ifdef DllDecl

```

```

#define DllDecl __declspec( dllimport )
#endif

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

    virtual PNEW_ORDER_DATA
    BuffAddr_NewOrder() = 0;
    virtual PPAYMENT_DATA
    BuffAddr_Payment() = 0;
    virtual PDELIVERY_DATA
    BuffAddr_Delivery() = 0;
    virtual PSTOCK_LEVEL_DATA
    BuffAddr_StockLevel() = 0;
    virtual PORDER_STATUS_DATA
    BuffAddr_OrderStatus() = 0;

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

```

db_dblib_dll\src\tpcc_dlib.cpp

```

/* FILE: TPC_C_DBLIB.CPP
 * Microsoft TPC-C Kit
Ver. 4.42.000
 * Copyright Microsoft,
2002
 * All Rights Reserved
 * Version 4.10.000
audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 * PURPOSE: Implements dblib calls for TPC-C txns.
 * Contact: Charles Levine (clevine@microsoft.com)
 *
 * Change history:
 * 4.42.000 - changed w_id fields from short to long
to support >32K warehouses
 * 4.20.000 - updated rev number to match kit
 * 4.10.001 - not deleting error class in catch handler
on deadlock retry;

```

```

 * not a functional bug,
 * but a memory leak - had to tweak some
 * declarations to compile with latest SDK; no functional change
 */

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

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

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

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

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

#define DEFCLPACKSIZE 4096

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

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

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

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

        case DLL_PROCESS_DETACH:
            dbexit(); // close
            all dblib structures/connections break;
    }
}

```

```

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

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

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

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

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

```

```

*/

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

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

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

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

    return 0;
}

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

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

```

```

}
return;

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
*
*/

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

    static SERRORMSG errorMsgs[] =
    {
        { ERR_WRONG_SP_VERSION,
        "Wrong version of stored procs on database server" },
        { ERR_INVALID_CUST,
        "Invalid Customer id,name." },
        { ERR_NO_SUCH_ORDER,
        "No
orders found for customer." },
        { ERR_RETRIED_TRANS,
        "Retries
before transaction succeeded." },
        { 0, "" }
    };

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

    for(i=0; errorMsgs[i].szMsg[0]; i++)
    {
        if ( m_erno == errorMsgs[i].iError )
            break;
    }

    if ( !errorMsgs[i].szMsg[0] )
        return szNotFound;
    else
        return errorMsgs[i].szMsg;
}

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

```

```

    LPCSTR szDatabase ) // name of database to
use
    return new CTPCC_DBLIB( szServer, szUser, szPassword,
szHost, szDatabase );
}

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

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

    m_MaxRetries = 10; // how many retries on
deadlock

    // increase max number of connections if getting close
    if ( dbgetmaxprocs() < (iConnectionCount+5) )
    {
        if ( dbsetmaxprocs(iConnectionCount+10) ==
FAIL )

        ThrowError(CDBLIBERR::eDbSetMaxProcs);
    }

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

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

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

    DBSETLUSER(login, szUser);
}

```

```

DBSETPWD(login, szPassword);
DBSETHOST(login, szHost);
DBSETLPACKET(login, (unsigned short)DEFCLPACKSIZE);
DBSETLVERSION(login, DBVER60); // use
dblib ver 6.0 client behavior

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

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

m_dbproc = dbopen(login, szServer);

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

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

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

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

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

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

DiscardNextResults(2);

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

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

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

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

```

```

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

DiscardNextRows(0);
DiscardNextResults(0);
}

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

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

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

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

void CTPCC_DBLIB::SetSqlError( int /*DBINT*/ msgno, int msgstate, int
severity, LPCSTR msgtext )

```

```

{
    if (m_SqlErr == NULL)
        m_SqlErr = new CSQLERR();

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

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

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

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

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

    throw pDbLibErr;
}

// Read and discard rows until no more. Throw an exception if number of
rows read doesn't

```

```

// match number of rows expected. The row count will be ignored if the
// expected count value
// passed in is negative. A typical use of this routine is to verify that there
// are no more
// rows to be read.
void CTPCC_DBLIB::DiscardNextRows(int iExpectedCount)
{
    int iRowsRead = 0;
    RETCODE rc;

    while (TRUE)
    {
        rc = dbnextrow(m_dbproc);
        if (rc == NO_MORE_ROWS)
            break;
        if (rc == FAIL)
        {
            if (iExpectedCount >= 0)

                ThrowError(CDBLIBERR::eDbNextRow);
            else
                break;
        }
        iRowsRead++;
    }

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

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

    while (TRUE)
    {
        rc = dbresults(m_dbproc);
        if (rc == NO_MORE_RESULTS)
            break;
        if (rc == FAIL)
        {
            if (iExpectedCount >= 0)

```

```

        ThrowError(CDBLIBERR::eDbResults);
    }

    DiscardNextRows(-1);
    iResultsRead++;
}

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

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

    ResetError();

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

            dbrpcparam(m_dbproc, NULL, 0,
                SQLINT4, -1, -1, (BYTE *) &m_txn.StockLevel.w_id); // @w_id
            int
            dbrpcparam(m_dbproc, NULL, 0,
                SQLINT1, -1, -1, (BYTE *) &m_txn.StockLevel.d_id); // @d_id
            tinyint
            dbrpcparam(m_dbproc, NULL, 0,
                SQLINT2, -1, -1, (BYTE *) &m_txn.StockLevel.threshold); // @threshold
            smallint

            if (dbrpcexec(m_dbproc) == FAIL)

                ThrowError(CDBLIBERR::eDbRpcExec);

            if (dbresults(m_dbproc) !=
                SUCCEED)

                ThrowError(CDBLIBERR::eDbResults);

            if (dbnextrow(m_dbproc) !=
                REG_ROW)

                ThrowError(CDBLIBERR::eDbNextRow);

            if (pData=dbdata(m_dbproc, 1))

```

```

            m_txn.StockLevel.low_stock = *((long *) pData);
            DiscardNextRows(0);
            DiscardNextResults(0);

            m_txn.StockLevel.exec_status_code
            = eOK;

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

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

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

    int iTryCount = 0;
    const BYTE *pData;

    ResetError();

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

```



```

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

        // check whether any order lines are
for a remote warehouse
        m_txn.NewOrder.o_all_local = 1;
        for (i = 0; i <
m_txn.NewOrder.o_ol_cnt; i++)
        {
                if
(m_txn.NewOrder.OL[i].ol_supply_w_id != m_txn.NewOrder.w_id)
                {

                        m_txn.NewOrder.o_all_local = 0; // at least one remote
warehouse
                                break;
                }
                dbrpcparam(m_dbproc, NULL, 0,
SQLINT1, -1, -1, (BYTE *) &m_txn.NewOrder.o_all_local);

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

                if (dbrpcexec(m_dbproc) == FAIL)

                        ThrowError(CDBLIBERR::eDbRpcExec);

                // Get order line results
                m_txn.NewOrder.total_amount = 0;
                for (i = 0;

i < m_txn.NewOrder.o_ol_cnt; i++)
                {

                        if

(dbresults(m_dbproc) != SUCCEED)

```

```

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

                ThrowError(CDBLIBERR::eWrongNumCols);

                if
(dbnextrow(m_dbproc) != REG_ROW)

                ThrowError(CDBLIBERR::eDbNextRow);

                if(pData=dbdata(m_dbproc, 1))

                UtilStrCpy(m_txn.NewOrder.OL[i].ol_i_name, pData,
dbdatlen(m_dbproc, 1));

                if(pData=dbdata(m_dbproc, 2))

                m_txn.NewOrder.OL[i].ol_stock = (*(DBSMALLINT *) pData);

                if(pData=dbdata(m_dbproc, 3))

                UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_generic, pData,
dbdatlen(m_dbproc, 3));

                if(pData=dbdata(m_dbproc, 4))

                dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,4),

                SQLFLT8,
                (BYTE *)&m_txn.NewOrder.OL[i].ol_i_price, 8);

                if(pData=dbdata(m_dbproc, 5))

                dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,5),

                SQLFLT8,
                (BYTE *)&m_txn.NewOrder.OL[i].ol_amount, 8);

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

                DiscardNextRows(0);

                // get remaining values for w_tax,
d_tax, o_id, c_last, c_discount, c_credit, o_entry_d, commit_flag

```

```

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

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

                if (dbnumcols(m_dbproc) != 8)

                ThrowError(CDBLIBERR::eWrongNumCols);

                if (pData=dbdata(m_dbproc, 1))

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

                if (pData=dbdata(m_dbproc, 2))

                dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,2), SQLFLT8, (BYTE
*)&m_txn.NewOrder.d_tax, 8);

                if (pData=dbdata(m_dbproc, 3))
                m_txn.NewOrder.o_id
                = (*(DBINT *) pData);

                if (pData=dbdata(m_dbproc, 4))

                UtilStrCpy(m_txn.NewOrder.c_last, pData, dbdatlen(m_dbproc,
4));

                if (pData=dbdata(m_dbproc, 5))
                dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,5), SQLFLT8, (BYTE
*)&m_txn.NewOrder.c_discount, 8);

                if (pData=dbdata(m_dbproc, 6))

                UtilStrCpy(m_txn.NewOrder.c_credit, pData,
dbdatlen(m_dbproc, 6));

                if (pData=dbdata(m_dbproc, 7))
                {
                        datetime =
*((DBDATETIME *) pData);

                        dbdatecrack(m_dbproc, &daterec, &datetime);

                        m_txn.NewOrder.o_entry_d.year = daterec.year;

                        m_txn.NewOrder.o_entry_d.month = daterec.month;

                        m_txn.NewOrder.o_entry_d.day = daterec.day;

```

```

m_txn.NewOrder.o_entry_d.hour = daterec.hour;

m_txn.NewOrder.o_entry_d.minute = daterec.minute;

m_txn.NewOrder.o_entry_d.second = daterec.second;
    }
    if (pData=dbdata(m_dbproc, 8))
        commit_flag =

(*(DBTINYINT *) pData);

        DiscardNextRows(0);
        DiscardNextResults(0);

        if (commit_flag == 1)
        {

            m_txn.NewOrder.total_amount *= ((1 +
m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));

            m_txn.NewOrder.exec_status_code = eOK;
        }
        else

            m_txn.NewOrder.exec_status_code = eInvalidItem;

            return;
        }
        catch (CSQLERR *e)
        {
            if ((e->m_msgno == 1205 ||
(e->m_msgno ==

iErrOleDbProvider &&

sErrTimeoutExpired) != NULL)) &&

            (+iTryCount <=

iMaxRetries))

        {
            // hit deadlock; backoff

            delete e;
            Sleep(10 * iTryCount);

        }
        else

            throw;

    }
    // while (TRUE)

// if (iTryCount)

```

```

// throw new
ETPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS, iTryCount);

void CTPCC_DBLIB::Payment()
{
    DBDATETIME datetime;
    DBDATERECEC daterec;

    int

    const BYTE *pData;

    ResetError();

    while (TRUE)
    {
        try
        {
            dbrpcinit(m_dbproc,

"tpcc_payment", 0);

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

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

            if (dbrpcexec(m_dbproc) == FAIL)

                ThrowError(CDBLIBERR::eDbRpcExec);

            if (dbresults(m_dbproc) !=
SUCCEED)

                ThrowError(CDBLIBERR::eDbResults);

            if (dbnextrow(m_dbproc) !=
REG_ROW)

```

```

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

            ThrowError(CDBLIBERR::eWrongNumCols);

            if (pData=dbdata(m_dbproc, 1))
                m_txn.Payment.c_id =

*((DBINT *) pData);

            if (pData=dbdata(m_dbproc, 2))

                UtilStrCpy(m_txn.Payment.c_last, pData, dbdatlen(m_dbproc,
2));

            if (pData=dbdata(m_dbproc, 3))
            {
                datetime =

*((DBDATETIME *) pData);

                dbdatecrack(m_dbproc, &daterec, &datetime);

                m_txn.Payment.h_date.year = daterec.year;

                m_txn.Payment.h_date.month = daterec.month;

                m_txn.Payment.h_date.day = daterec.day;

                m_txn.Payment.h_date.hour = daterec.hour;

                m_txn.Payment.h_date.minute = daterec.minute;

                m_txn.Payment.h_date.second = daterec.second;
            }
            if (pData=dbdata(m_dbproc, 4))

                UtilStrCpy(m_txn.Payment.w_street_1, pData,
dbdatlen(m_dbproc, 4));

            if (pData=dbdata(m_dbproc, 5))

                UtilStrCpy(m_txn.Payment.w_street_2, pData,
dbdatlen(m_dbproc, 5));

            if (pData=dbdata(m_dbproc, 6))

                UtilStrCpy(m_txn.Payment.w_city, pData, dbdatlen(m_dbproc,
6));

            if (pData=dbdata(m_dbproc, 7))

                UtilStrCpy(m_txn.Payment.w_state, pData, dbdatlen(m_dbproc,
7));

            if (pData=dbdata(m_dbproc, 8))

                UtilStrCpy(m_txn.Payment.w_zip, pData, dbdatlen(m_dbproc,
8));

            if (pData=dbdata(m_dbproc, 9))

```

```

        UtilStrCpy(m_txn.Payment.d_street_1, pData,
dbdatlen(m_dbproc, 9));
        if (pData=dbdata(m_dbproc, 10))

        UtilStrCpy(m_txn.Payment.d_street_2, pData,
dbdatlen(m_dbproc, 10));
        if (pData=dbdata(m_dbproc, 11))

        UtilStrCpy(m_txn.Payment.d_city, pData, dbdatlen(m_dbproc,
11));
        if (pData=dbdata(m_dbproc, 12))

        UtilStrCpy(m_txn.Payment.d_state, pData, dbdatlen(m_dbproc,
12));
        if (pData=dbdata(m_dbproc, 13))

        UtilStrCpy(m_txn.Payment.d_zip, pData, dbdatlen(m_dbproc,
13));
        if (pData=dbdata(m_dbproc, 14))

        UtilStrCpy(m_txn.Payment.c_first, pData, dbdatlen(m_dbproc,
14));
        if (pData=dbdata(m_dbproc, 15))

        UtilStrCpy(m_txn.Payment.c_middle, pData,
dbdatlen(m_dbproc, 15));
        if (pData=dbdata(m_dbproc, 16))

        UtilStrCpy(m_txn.Payment.c_street_1, pData,
dbdatlen(m_dbproc, 16));
        if (pData=dbdata(m_dbproc, 17))

        UtilStrCpy(m_txn.Payment.c_street_2, pData,
dbdatlen(m_dbproc, 17));
        if (pData=dbdata(m_dbproc, 18))

        UtilStrCpy(m_txn.Payment.c_city, pData, dbdatlen(m_dbproc,
18));
        if (pData=dbdata(m_dbproc, 19))

        UtilStrCpy(m_txn.Payment.c_state, pData, dbdatlen(m_dbproc,
19));
        if (pData=dbdata(m_dbproc, 20))

        UtilStrCpy(m_txn.Payment.c_zip, pData, dbdatlen(m_dbproc,
20));
        if (pData=dbdata(m_dbproc, 21))

        UtilStrCpy(m_txn.Payment.c_phone, pData,
dbdatlen(m_dbproc, 21));
        if (pData=dbdata(m_dbproc, 22))

```

```

        {
            datetime =
*((DBDATETIME *) pData);

            dbdatecrack(m_dbproc, &daterec, &datetime);

            m_txn.Payment.c_since.year = daterec.year;

            m_txn.Payment.c_since.month = daterec.month;

            m_txn.Payment.c_since.day = daterec.day;

            m_txn.Payment.c_since.hour = daterec.hour;

            m_txn.Payment.c_since.minute = daterec.minute;

            m_txn.Payment.c_since.second = daterec.second;
        }
        if(pData=dbdata(m_dbproc, 23))

        UtilStrCpy(m_txn.Payment.c_credit, pData, dbdatlen(m_dbproc,
23));
        if(pData=dbdata(m_dbproc, 24))
            dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,24), SQLFLT8, (BYTE
*)&m_txn.Payment.c_credit_lim, 8);
        if(pData=dbdata(m_dbproc, 25))
            dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,25), SQLFLT8, (BYTE
*)&m_txn.Payment.c_discount, 8);
        if(pData=dbdata(m_dbproc, 26))
            dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,26), SQLFLT8, (BYTE
*)&m_txn.Payment.c_balance, 8);
        if(pData=dbdata(m_dbproc, 27))

        UtilStrCpy(m_txn.Payment.c_data, pData, dbdatlen(m_dbproc,
27));

        DiscardNextRows(0);
        DiscardNextResults(0);

        if (m_txn.Payment.c_id == 0)
            throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
        else

            m_txn.Payment.exec_status_code = eOK;

        return;
    }
    catch (CSQLERR *e)
    {

```

```

        if ((e->m_msgcode != 1205) !=
iErrOleDbProvider &&
sErrTimeoutExpired) != NULL)) &&
iMaxRetries))
        {
            // hit deadlock; backoff
            delete e;
            Sleep(10 * iTryCount);
        }
        else
            throw;
    }
    // while (TRUE)
    // if (iTryCount)
    //     throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS, iTryCount);
}

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

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

    ResetError();

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

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

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

```

```

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

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

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

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

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

            if(pData=dbdata(m_dbproc, 1))
                m_txn.OrderStatus.OL[i].ol_supply_w_id = (*(DBSMALLINT *)
pData);

            if(pData=dbdata(m_dbproc, 2))
                m_txn.OrderStatus.OL[i].ol_i_id = (*(DBINT *) pData);

            if(pData=dbdata(m_dbproc, 3))
                m_txn.OrderStatus.OL[i].ol_quantity = (*(DBSMALLINT *)
pData);

```

```

            if(pData=dbdata(m_dbproc, 4))
                dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,4),
                SQLFLT8, (BYTE
*)&m_txn.OrderStatus.OL[i].ol_amount, 8);

            if(pData=dbdata(m_dbproc, 5))
                {
                    datetime
                    = *((DBDATETIME *) pData);
                    dbdatecrack(m_dbproc, &daterec, &datetime);
                    m_txn.OrderStatus.OL[i].ol_delivery_d.year = daterec.year;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.month =
                    daterec.month;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.day = daterec.day;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.hour = daterec.hour;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.minute =
                    daterec.minute;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.second =
                    daterec.second;
                }
                i++;
            }
            m_txn.OrderStatus.o_ol_cnt = i;

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

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

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

            if(pData=dbdata(m_dbproc, 1))
                m_txn.OrderStatus.c_id = (*(DBINT *) pData);
            if(pData=dbdata(m_dbproc, 2))

```

```

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

                DiscardNextRows(0);
                DiscardNextResults(0);

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

```

```

        else
            m_txn.OrderStatus.exec_status_code = eOK;

        return;
    } catch (CSQLERR *e)
    {
        if ((e->m_msgno == 1205 ||
            (e->m_msgno ==

iErrOleDbProvider &&
sErrTimeoutExpired) != NULL)) &&
            iMaxRetries))
        {
            // hit deadlock; backoff
            for increasingly longer period
                delete e;
                Sleep(10 * iTryCount);
            }
        else
            throw;
    }
} // while (TRUE)

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

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

    ResetError();

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

0);

            dbrpcparam(m_dbproc, NULL, 0,
SQLINT4, -1, -1, (BYTE *) &m_txn.Delivery.w_id);
            dbrpcparam(m_dbproc, NULL, 0,
SQLINT1, -1, -1, (BYTE *) &m_txn.Delivery.o_carrier_id);

```

```

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

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

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

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

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

        DiscardNextRows(0);
        DiscardNextResults(0);

        m_txn.Delivery.exec_status_code =
eOK;

        return;
    } catch (CSQLERR *e)
    {
        if ((e->m_msgno == 1205 ||
            (e->m_msgno ==

iErrOleDbProvider &&
sErrTimeoutExpired) != NULL)) &&
            iMaxRetries))
        {
            // hit deadlock; backoff
            for increasingly longer period
                delete e;
                Sleep(10 * iTryCount);
            }
        else
            throw;
    }
} // while (TRUE)

// if (iTryCount)

```

```

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

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

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

    return;
}

db_odbc.dll tpcc_odbc.dll

/* FILE: TPCC_ODBC.CPP
 *
 * Microsoft TPC-C Kit
 * Ver. 4.42.000
 * Copyright Microsoft,
 * 2002
 * All Rights Reserved
 *
 * Version 4.10.000
 * audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 * PURPOSE: Implements ODBC calls for TPC-C txns.
 * Contact: Charles Levine (clevine@microsoft.com)
 *
 * Change history:
 * 4.42.000 - changed w_id fields from short to long
to support >32K warehouses
 * 4.20.000 - updated rev number to match kit
 * 4.10.001 - not deleting error class in catch handler
on deadlock retry;
 * not a functional bug,
but a memory leak
 */

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

```

```

#include <assert.h>

#define DBNTWIN32
#include <sqltypes.h>
#include <sql.h>
#include <sqltext.h>
#include <odbcss.h>

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

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

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

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

const iMaxRetries = 10;           // how many retries on deadlock

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

static SQLHENV henv = SQL_NULL_HENV;
// ODBC environment handle

BOOL APIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call,
LPVOID lpReserved)
{
    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:
            DisableThreadLibraryCalls(hModule);
            if (
SQLAllocHandleStd(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv) !=
SQL_SUCCESS )
                return FALSE;
            break;

        case DLL_PROCESS_DETACH:
            if (henv != NULL)
                SQLFreeEnv(henv);
            break;

        default:
            /* nothing */;
    }
}

```

```

}
return TRUE;

/* FUNCTION: CTPCC_ODBC_ERR::ErrorText
*
*/

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

    static SERRORMSG errorMsgs[] =
    {
        { ERR_WRONG_SP_VERSION,
        "Wrong version of stored procs on database server" },
        { ERR_INVALID_CUST,
        "Invalid Customer id,name." },
        { ERR_NO_SUCH_ORDER,
        "No orders found for customer." },
        { ERR_RETRIED_TRANS,
        "Retries before transaction succeeded." },
        { 0,
        "" }
    };

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

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

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_ODBC* CTPCC_ODBC_new(
    LPCSTR szServer,           // name of SQL server
    LPCSTR szUser,             // user
    LPCSTR szPassword,         // password for login
    LPCSTR szHost,             // not
    LPCSTR szDatabase )        // name of database to
use

```

```

{
    return new CTPCC_ODBC( szServer, szUser, szPassword,
szHost, szDatabase );
}

CTPCC_ODBC::CTPCC_ODBC (
    LPCSTR szServer,
    // name of SQL server
    LPCSTR szUser,
    // user name for login
    LPCSTR szPassword,
    // password for login
    LPCSTR szHost,
    // not used
    LPCSTR szDatabase
    // name of database to use
)
{
    RETCODE rc;

    // initialization
    m_hdbc = SQL_NULL_HDBC;
    m_hstmt = SQL_NULL_HSTMT;

    m_hstmtNewOrder = SQL_NULL_HSTMT;
    m_hstmtPayment = SQL_NULL_HSTMT;
    m_hstmtDelivery = SQL_NULL_HSTMT;
    m_hstmtOrderStatus = SQL_NULL_HSTMT;
    m_hstmtStockLevel = SQL_NULL_HSTMT;

    m_descNewOrderCols1 = SQL_NULL_HDESC;
    m_descNewOrderCols2 = SQL_NULL_HDESC;
    m_descOrderStatusCols1 = SQL_NULL_HDESC;
    m_descOrderStatusCols2 = SQL_NULL_HDESC;

    if ( SQLAllocHandle(SQL_HANDLE_DBC, henv, &m_hdbc) !=
SQL_SUCCESS )
        ThrowError(CODBCERR::eAllocHandle);

    if ( SQLSetConnectOption(m_hdbc, SQL_PACKET_SIZE, 4096)
!= SQL_SUCCESS )
        ThrowError(CODBCERR::eConnOption);

    {
        char
szConnectStr[256];
        char
szOutStr[1024];
        SQLSMALLINT
iOutStrLen;
    }
}

```

```

        sprintf( szConnectStr, "DRIVER=SQL
Server;SERVER=%s;UID=%s;PWD=%s;DATABASE=%s",
                szServer, szUser, szPassword,
                szDatabase );

        rc = SQLDriverConnect(m_hdbc, NULL,
(SQLCHAR*)szConnectStr, sizeof(szConnectStr),
                (SQLCHAR*)szOutStr,
                sizeof(szOutStr), &iOutStrLen, SQL_DRIVER_NOPROMPT );

        if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)
                ThrowError(CODBCERR::eConnect);
    }

    if (SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmt)
!= SQL_SUCCESS)
        ThrowError(CODBCERR::eAllocHandle);

    {
        char                buffer[128];

        // set some options affecting connection behavior
        strcpy(buffer, "set nocount on set XACT_ABORT
ON");

        rc = SQLExecDirect(m_hstmt, (unsigned char
*)buffer, SQL_NTS);

        if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)
            ThrowError(CODBCERR::eExecDirect);

        // verify that version of stored procs on server is
correct
        char db_sp_version[10];
        strcpy(buffer, "{call tpcc_version}");
        rc = SQLExecDirect(m_hstmt, (unsigned char
*)buffer, SQL_NTS);

        if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)
            ThrowError(CODBCERR::eExecDirect);

        if ( SQLBindCol(m_hstmt, 1, SQL_C_CHAR,
&db_sp_version, sizeof(db_sp_version), NULL) != SQL_SUCCESS )
            ThrowError(CODBCERR::eBindCol);

        if ( SQLFetch(m_hstmt) == SQL_ERROR )
            ThrowError(CODBCERR::eFetch);

        if (strcmp(db_sp_version, sVersion))
            throw new CTPCC_ODBC_ERR(
CTPCC_ODBC_ERR::ERR_WRONG_SP_VERSION );
    }

```

```

    }
    SQLFreeHandle(SQL_HANDLE_STMT, m_hstmt);

    // Bind parameters for each of the transactions
    InitNewOrderParams();
    InitPaymentParams();
    InitOrderStatusParams();
    InitDeliveryParams();
    InitStockLevelParams();
}

CTPCC_ODBC::~CTPCC_ODBC( void )
{
    // note: descriptors are automatically released when the
connection is dropped
    SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtNewOrder);
    SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtPayment);
    SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtDelivery);
    SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtOrderStatus);
    SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtStockLevel);

    SQLDisconnect(m_hdbc);
    SQLFreeHandle(SQL_HANDLE_DBC, m_hdbc);
}

void CTPCC_ODBC::ThrowError( CODBCERR::ACTION eAction )
{
    RETCODE                rc;
    SDWORD                INativeError;
    char                szState[6];
    char                szMsg[SQL_MAX_MESSAGE_LENGTH];
    char                szTmp[6*SQL_MAX_MESSAGE_LENGTH];
    CODBCERR    *pODBCErr;

    // not
    allocated until needed (maybe never)

    pODBCErr = new CODBCERR();

    pODBCErr->m_NativeError = 0;
    pODBCErr->m_eAction = eAction;
    pODBCErr->m_bDeadLock = FALSE;

    szTmp[0] = 0;
    while (TRUE)
    {
        rc = SQLError(henv, m_hdbc, m_hstmt, (BYTE
*)&szState, &INativeError,
                (BYTE
*)&szMsg, sizeof(szMsg), NULL);

        if (rc == SQL_NO_DATA)
            break;

        // check for deadlock
    }
}

```

```

        if (INativeError == 1205 || (INativeError ==
strchr(szMsg, sErrTimeoutExpired) !=
NULL))
            pODBCErr->m_bDeadLock = TRUE;

        // capture the (first) database error
        if (pODBCErr->m_NativeError == 0 &&
INativeError != 0)
            pODBCErr->m_NativeError =
INativeError;

        // quit if there isn't enough room to concatenate
        if ( (strlen(szMsg) + 2) > (sizeof(szTmp) -
strlen(szTmp)) )
            break;

        // include line break after first error msg
        if (szTmp[0] != 0)
            strcat( szTmp, "\n");
        strcat( szTmp, szMsg );
    }

    if (pODBCErr->m_odbcerrstr != NULL)
    {
        delete [] pODBCErr->m_odbcerrstr;
        pODBCErr->m_odbcerrstr = NULL;
    }

    if (strlen(szTmp) > 0)
    {
        pODBCErr->m_odbcerrstr = new char[
strlen(szTmp)+1 ];
        strcpy( pODBCErr->m_odbcerrstr, szTmp );
    }

    SQLFreeStmt(m_hstmt, SQL_CLOSE);
    throw pODBCErr;
}

void CTPCC_ODBC::InitStockLevelParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtStockLevel) != SQL_SUCCESS )
        ThrowError(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtStockLevel;

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SLONG, SQL_INTEGER, 0, 0, &m_txn.StockLevel.w_id, 0, NULL) !=
SQL_SUCCESS

```

```

        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.StockLevel.d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
&m_txn.StockLevel.threshold, 0, NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindParam);

    if ( SQLBindCol(m_hstmt, 1, SQL_C_SLONG,
&m_txn.StockLevel.low_stock, 0, NULL) != SQL_SUCCESS )
        ThrowError(CODBCERR::eBindCol);
}

void CTPCC_ODBC::StockLevel()
{
    RETCODE          rc;
    int               iTryCount = 0;

    m_hstmt = m_hstmtStockLevel;

    while (TRUE)
    {
        try
        {
            rc = SQLExecDirectW(m_hstmt,
(SQLWCHAR*)"L\"call tpcc_stocklevel(?,?,?)\"", SQL_NTS);
            if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

                ThrowError(CODBCERR::eExecDirect);

            if ( SQLFetch(m_hstmt) ==
SQL_ERROR )

                ThrowError(CODBCERR::eFetch);

            SQLFreeStmt(m_hstmt,

                m_txn.StockLevel.exec_status_code

            );

            break;
        }
        catch (CODBCERR *e)
        {
            if ((!e->m_bDeadLock) ||
(++iTryCount > iMaxRetries))

                throw;

            // hit deadlock; backoff for
increasingly longer period

```

```

        }
    }

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

void CTPCC_ODBC::InitNewOrderParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtNewOrder) != SQL_SUCCESS
        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderCols1) != SQL_SUCCESS
        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderCols2) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtNewOrder;

    if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descNewOrderCols1, SQL_IS_POINTER ) != SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SLONG, SQL_INTEGER, 0, 0, &m_txn.NewOrder.w_id, 0, NULL) !=
SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.NewOrder.d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.c_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.NewOrder.o_ol_cnt, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.NewOrder.o_all_local, 0, NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindParam);

    for (int j=0; j<MAX_OL_NEW_ORDER_ITEMS; j++)
    {
        if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.OL[j].ol_i_id, 0, NULL) != SQL_SUCCESS
            || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.OL[j].ol_supply_w_id, 0, NULL) != SQL_SUCCESS

```

```

        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
&m_txn.NewOrder.OL[j].ol_quantity, 0, NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindParam);
    }

    // set the bind offset pointer
    if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_ROW_BIND_OFFSET_PTR, &m_BindOffset, SQL_IS_POINTER )
!= SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);

    i = 0;
    if ( SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.OL[0].ol_i_name,
sizeof(m_txn.NewOrder.OL[0].ol_i_name), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.NewOrder.OL[0].ol_stock, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.OL[0].ol_brand_generic,
sizeof(m_txn.NewOrder.OL[0].ol_brand_generic), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.OL[0].ol_i_price, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.OL[0].ol_amount, 0, NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindCol);

    // associate the column bindings for the second result set
    if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descNewOrderCols2, SQL_IS_POINTER ) != SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);

    i = 0;
    if ( SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.w_tax, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.d_tax, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.NewOrder.o_id, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.c_last, sizeof(m_txn.NewOrder.c_last), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.c_discount, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.c_credit, sizeof(m_txn.NewOrder.c_credit), NULL) !=
SQL_SUCCESS
    )
        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.NewOrder.o_entry_d, 0, NULL) !=
SQL_SUCCESS

```



```

        || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_no_commit_flag, 0, NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindCol);
}

void CTPCC_ODBC::NewOrder()
{
    int
    i;
    RETCODE          rc;
    int
    iTryCount = 0;

    // 0

    //
    012345678901234567890123456789
    wchar_t
    szSqlTemplate[] = L"{call tpcc_neworder(?,?,?,?,,"

    L"?,?,?,?,?,?,?,?,?,?,?,?,,"

    L"?,?,?,?,?,?,?,?,?,?,?,?,,"

    L"?,?,?,?,?,?,?,?,?,?,?,?,?)}";

    m_hstmt = m_hstmtNewOrder;

    // associate the parameter and column bindings for this
transaction
    if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descNewOrderCols1, SQL_IS_POINTER ) != SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);

    // clip statement buffer based on number of parameters
    // fixed part is 29 chars and variable part is 6 chars per line
item
    i = 29 + m_txn.NewOrder.o_ol_cnt*6;
    wcsncpy( &szSqlTemplate[i], L" )" );

    // check whether any order lines are for a remote warehouse
    m_txn.NewOrder.o_all_local = 1;
    for ( i = 0; i < m_txn.NewOrder.o_ol_cnt; i++)
    {
        if (m_txn.NewOrder.OL[i].ol_supply_w_id !=
m_txn.NewOrder.w_id)
    {

```

```

        at least one remote warehouse
        }
    }

    while (TRUE)
    {
        try
        {
            m_BindOffset = 0;
            rc = SQLExecDirectW(m_hstmt,
(SQLWCHAR*)szSqlTemplate, SQL_NTS);
            if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)
                ThrowError(CODBCERR::eExecDirect);

            // Get order line results
            m_txn.NewOrder.total_amount = 0;
            for (i = 0;
i < m_txn.NewOrder.o_ol_cnt; i++)
            {
                // set the bind offset
                m_BindOffset = i *
                sizeof(m_txn.NewOrder.OL[0]);

                if ( SQLFetch(m_hstmt)
                == SQL_ERROR)
                    ThrowError(CODBCERR::eFetch);

                // move to the next
                if (
SQLMoreResults(m_hstmt) == SQL_ERROR )
                    ThrowError(CODBCERR::eMoreResults);

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

            // associate the column bindings for
the second result set
            if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descNewOrderCols2, SQL_IS_POINTER ) !=
SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);

```

```

        if ( SQLFetch(m_hstmt) ==
SQL_ERROR)
            ThrowError(CODBCERR::eFetch);

        SQLFreeStmt(m_hstmt,
SQL_CLOSE);

        if (m_no_commit_flag == 1)
        {
            m_txn.NewOrder.total_amount *= ((1 +
m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));

            m_txn.NewOrder.exec_status_code = eOK;
        }
        else
            m_txn.NewOrder.exec_status_code = eInvalidItem;

        break;
    }
    catch (CODBCERR *e)
    {
        if ((!e->m_bDeadLock) ||
(++iTryCount > iMaxRetries))
            throw;

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

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

void CTPCC_ODBC::InitPaymentParams()
{
    {
        if ( SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtPayment) != SQL_SUCCESS )
            ThrowError(CODBCERR::eAllocHandle);

        m_hstmt = m_hstmtPayment;

        int i = 0;
        if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SLONG, SQL_INTEGER, 0, 0, &m_txn.Payment.w_id, 0, NULL) !=
SQL_SUCCESS

```

```

        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.Payment.c_w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_DOUBLE, SQL_NUMERIC, 6, 2,
&m_txn.Payment.h_amount, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.Payment.d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.Payment.c_d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.Payment.c_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
sizeof(m_txn.Payment.c_last), 0, &m_txn.Payment.c_last,
sizeof(m_txn.Payment.c_last), NULL) != SQL_SUCCESS
    )
    ThrowError(CODBCERR::eBindParam);

    i = 0;
    if ( SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.Payment.c_id,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.c_last), NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.Payment.h_date,
0, NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.w_street_1),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.w_street_2),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.w_city),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.w_state),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.w_zip), NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.d_street_1),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.d_street_2),
NULL) != SQL_SUCCESS

```

```

        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.d_city),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.d_state),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.d_zip), NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.c_first),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.c_middle),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.c_street_1),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.c_street_2),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.c_city), NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.c_state),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.c_zip), NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.c_phone),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.Payment.c_since,
0, NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.c_credit), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.Payment.c_credit_lim, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.Payment.c_discount,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.Payment.c_balance,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
sizeof(m_txn.Payment.c_data),
NULL) != SQL_SUCCESS
    )
    ThrowError(CODBCERR::eBindCol);
}

void CTPCC_ODBC::Payment()

```

```

{
    RETCODE          rc;
    int               iTryCount = 0;

    m_hstmt = m_hstmtPayment;

    if (m_txn.Payment.c_id != 0)
        m_txn.Payment.c_last[0] = 0;

    while (TRUE)
    {
        try
        {
            rc = SQLEXPDirectW(m_hstmt,
(SQLWCHAR*)"L" + "call tpcc_payment(?,?,?,?,?)", SQL_NTS);
            if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

                ThrowError(CODBCERR::eExecDirect);

            if ( SQLFetch(m_hstmt) ==
SQL_ERROR)

                ThrowError(CODBCERR::eFetch);

            SQLFreeStmt(m_hstmt,
SQL_CLOSE);

            if (m_txn.Payment.c_id == 0)
                throw new
CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_INVALID_CUST );
            else

                m_txn.Payment.exec_status_code = eOK;

            break;
        }
        catch (CODBCERR *e)
        {
            if ((!e->m_bDeadLock) ||
(++iTryCount > iMaxRetries))

                throw;

            // hit deadlock; backoff for
            increasingly longer period

            delete e;
            Sleep(10 * iTryCount);
        }
    }

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

```

```

}

void CTPCC_ODBC::InitOrderStatusParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtOrderStatus) != SQL_SUCCESS
        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols1) != SQL_SUCCESS
        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols2) != SQL_SUCCESS
        )
        ThrowError(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtOrderStatus;

    if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols1, SQL_IS_POINTER ) != SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SLONG, SQL_INTEGER, 0, 0, &m_txn.OrderStatus.w_id, 0, NULL) !=
SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.OrderStatus.d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.OrderStatus.c_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
sizeof(m_txn.OrderStatus.c_last), 0, &m_txn.OrderStatus.c_last,
sizeof(m_txn.OrderStatus.c_last), NULL) != SQL_SUCCESS
        )
        ThrowError(CODBCERR::eBindParam);

    // configure block cursor
    if ( SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_BIND_TYPE,
(SQLPOINTER)sizeof(m_txn.OrderStatus.OL[0]), 0) != SQL_SUCCESS
        || SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROWS_FETCHED_PTR, &m_RowsFetched, 0) != SQL_SUCCESS
        )
        ThrowError(CODBCERR::eSetStmtAttr);

    i = 0;
    if ( SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.OL[0].ol_supply_w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.OL[0].ol_i_id, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.OrderStatus.OL[0].ol_quantity, 0, NULL) != SQL_SUCCESS

```

```

        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.OrderStatus.OL[0].ol_amount, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_SUCCESS,
SQL_C_TYPE_TIMESTAMP, &m_txn.OrderStatus.OL[0].ol_delivery_d, 0,
NULL) != SQL_SUCCESS
        )
        ThrowError(CODBCERR::eBindCol);

    if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols2, SQL_IS_POINTER ) != SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);

    i = 0;
    if ( SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.c_id, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_last, sizeof(m_txn.OrderStatus.c_last), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_first, sizeof(m_txn.OrderStatus.c_first), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_middle, sizeof(m_txn.OrderStatus.c_middle), NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.OrderStatus.o_entry_d, 0, NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.OrderStatus.o_carrier_id, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.OrderStatus.c_balance, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.o_id, 0, NULL) != SQL_SUCCESS
        )
        ThrowError(CODBCERR::eBindCol);
}

void CTPCC_ODBC::OrderStatus()
{
    int
        iTryCount = 0;

    RETCODE
        rc;

    m_hstmt = m_hstmtOrderStatus;

    if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols1, SQL_IS_POINTER ) != SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);

    if (m_txn.OrderStatus.c_id != 0)
        m_txn.OrderStatus.c_last[0] = 0;
}

```

```

while (TRUE)
    try
    {
        // configure block cursor
        if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_ARRAY_SIZE, (SQLPOINTER)1, 0) != SQL_SUCCESS )

            ThrowError(CODBCERR::eSetStmtAttr);

        rc = SQLExecDirectW(m_hstmt,
(SQLWCHAR*)"L"{call tpcc_orderstatus{?,?,?}}", SQL_NTS);
        if ( ((rc ==
SQL_SUCCESS_WITH_INFO) && (m_RowsFetched != 0)) || (rc ==
SQL_ERROR) )

            ThrowError(CODBCERR::eExecDirect);

        // configure block cursor
        if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)MAX_OL_ORDER_STATUS_ITEMS, 0) != SQL_SUCCESS )

            ThrowError(CODBCERR::eSetStmtAttr);

        rc = SQLFetchScroll( m_hstmt,
SQL_FETCH_NEXT, 0 );

        if ( ((rc ==
SQL_SUCCESS_WITH_INFO) && (m_RowsFetched != 0)) || (rc ==
SQL_ERROR) )

            ThrowError(CODBCERR::eFetchScroll);

        m_txn.OrderStatus.o_ol_cnt =
(short)m_RowsFetched;

        if (m_txn.OrderStatus.o_ol_cnt != 0)
        {
            if ( SQLSetStmtAttrW(
m_hstmt, SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols2,
SQL_IS_POINTER ) != SQL_SUCCESS )

                ThrowError(CODBCERR::eSetStmtAttr);

            if (
SQLMoreResults(m_hstmt) == SQL_ERROR )

                ThrowError(CODBCERR::eMoreResults);

            if ( (rc =
SQLFetch(m_hstmt)) == SQL_ERROR)

                ThrowError(CODBCERR::eFetch);

```

```

    }

    SQLFreeStmt(m_hstmt,

SQL_CLOSE);

    if (m_txn.OrderStatus.o_ol_cnt ==

0)

        throw new

CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_NO_SUCH_ORDER );

    else if (m_txn.OrderStatus.c_id == 0

&& m_txn.OrderStatus.c_last[0] == 0)

        throw new

CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_INVALID_CUST );

    else

        m_txn.OrderStatus.exec_status_code = eOK;

        break;

    }

    catch (CDBCERR *e)

    {

        if ((1e->m_bDeadLock) ||

(++iTryCount > iMaxRetries))

            throw;

        // hit deadlock; backoff for

        delete e;

        Sleep(10 * iTryCount);

    }

}

// if (iTryCount)

// throw new

CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS, iTryCount);

}

void CTPCC_ODBC::InitDeliveryParams()

{

    if ( SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,

&m_hstmtDelivery) != SQL_SUCCESS )

        ThrowError(CDBCERR::eAllocHandle);

    m_hstmt = m_hstmtDelivery;

    int i = 0;

    if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,

SQL_C_SLONG, SQL_INTEGER, 0, 0, &m_txn.Delivery.w_id, 0, NULL) !=

SQL_SUCCESS

        || SQLBindParameter(m_hstmt, ++i,

SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,

&m_txn.Delivery.o_carrier_id, 0, NULL) != SQL_SUCCESS


```

```

        }

        ThrowError(CDBCERR::eBindParam);

        for (i=0;i<10;i++)

        {

            if ( SQLBindCol(m_hstmt, (UWORD)(i+1),

SQL_C_SLONG, &m_txn.Delivery.o_id[i], 0, NULL) != SQL_SUCCESS )

                ThrowError(CDBCERR::eBindCol);

        }

    }

    void CTPCC_ODBC::Delivery()

    {

        RETCODE rc;

        int iTryCount = 0;

        m_hstmt = m_hstmtDelivery;

        while (TRUE)

        {

            try

            {

                rc = SQLExecDirectW(m_hstmt,

(SQLWCHAR*)L"{call tpcc_delivery(?,?)}", SQL_NTS);

                if (rc != SQL_SUCCESS && rc !=

SQL_SUCCESS_WITH_INFO)

                    ThrowError(CDBCERR::eExecDirect);

                if ( SQLFetch(m_hstmt) ==

SQL_ERROR )

                    ThrowError(CDBCERR::eFetch);

                SQLFreeStmt(m_hstmt,

SQL_CLOSE);

                m_txn.Delivery.exec_status_code =

eOK;

                break;

            }

            catch (CDBCERR *e)

            {

                if ((1e->m_bDeadLock) ||

(++iTryCount > iMaxRetries))

                    throw;

                // hit deadlock; backoff for

                delete e;

                Sleep(10 * iTryCount);

            }

        }

    }

    // if (iTryCount)


```

```

// throw new

CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS, iTryCount);


```

db_odbc.dll tpcc_odbc.h

```

/* FILE: TPCC_ODBC.H

* Microsoft TPC-C Kit

Ver. 4.20.000

* Copyright Microsoft,

1999

* All Rights Reserved

* Version 4.10.000

audited by Richard Gimarc, Performance Metrics, 3/17/99

* PURPOSE: Header file for TPC-C txn class implementation.

* Change history:

* 4.20.000 - updated rev number to match kit

*/

#pragma once

// need to declare functions for import, unless define has already been

created

// by the DLL's .cpp module for export.

#ifndef DllDecl

#define DllDecl __declspec( dllimport )

#endif

class CDBCERR : public CBaseErr

{

public:

    enum ACTION

    {

        eNone,

        eUnknown,

        eAllocConn,

        // error from SQLAllocConnect

        eAllocHandle, // error

        from SQLAllocHandle

        eConnOption,

        // error from SQLSetConnectOption

        eConnect,

        // error from SQLConnect

    }


```

```

        eAllocStmt,
        // error from SQLAllocStmt
        eExecDirect, // error
    from SQLExecDirect
        eBindParam,
        // error from SQLBindParameter
        eBindCol,
        // error from SQLBindCol
        eFetch,
        // error from SQLFetch
        eFetchScroll, // error
    from SQLFetchScroll
        eMoreResults,
        // error from SQLMoreResults
        ePrepare,
        // error from SQLPrepare
        eExecute,
        // error from SQLExecute
        eSetEnvAttr, // error
    from SQLSetEnvAttr
        eSetStmtAttr // error
    from SQLSetStmtAttr
    };

    CODBECERR(void)
    {
        m_eAction = eNone;
        m_NativeError = 0;
        m_bDeadLock = FALSE;
        m_odbcerrstr = NULL;
    };

    ~CODBECERR()
    {
        if (m_odbcerrstr != NULL)
            delete [] m_odbcerrstr;
    };

    ACTION m_eAction;
    int m_NativeError;
    BOOL m_bDeadLock;
    char *m_odbcerrstr;

    int ErrorType() {return ERR_TYPE_ODBC;};
    int ErrorNum() {return m_NativeError;};
    char *ErrorText() {return m_odbcerrstr;};

};

class CTPCC_ODBC_ERR : public CBaseErr
{

```

```

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

    CTPCC_ODBC_ERR( int iErr ) { m_erno = iErr;
    m_iTryCount = 0; };

    CTPCC_ODBC_ERR( int iErr, int iTryCount ) {
    m_erno = iErr; m_iTryCount = iTryCount; };

    int m_erno;
    int m_iTryCount;

    int ErrorType() {return ERR_TYPE_TPCC_ODBC;};
    int ErrorNum() {return m_erno;};

    char *ErrorText();

};

class DllDecl CTPCC_ODBC : public CTPCC_BASE
{
    private:
        // declare variables and private functions here...
        BOOL m_bDeadlock;
        // transaction was selected as deadlock victim
        int m_MaxRetries; // retry count on
        deadlock

        SQLHENV m_henv;
        SQLHDBC // ODBC environment handle
        m_hdbc;
        SQLHSTMT m_hstmt;
        // the current hstmt

        SQLHSTMT m_hstmtNewOrder;
        SQLHSTMT m_hstmtPayment;
        SQLHSTMT m_hstmtDelivery;
        SQLHSTMT m_hstmtOrderStatus;
        SQLHSTMT m_hstmtStockLevel;

        SQLHDESC m_descNewOrderCols1;
        SQLHDESC m_descNewOrderCols2;
        SQLHDESC m_descOrderStatusCols1;
        SQLHDESC m_descOrderStatusCols2;

```

```

        // new-order specific fields
        SQLINTEGER m_BindOffset;
        SQLINTEGER m_RowsFetched;
        int m_no_commit_flag;

        void ThrowError( CODBECERR::ACTION eAction );

        void InitNewOrderParams();
        void InitPaymentParams();
        void InitDeliveryParams();
        void InitStockLevelParams();
        void InitOrderStatusParams();

        union
        {
            NEW_ORDER_DATA
            Payment_DATA
            DELIVERY_DATA
            STOCK_LEVEL_DATA
            ORDER_STATUS_DATA
        } m_txn;

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

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

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

```

```

};

// wrapper routine for class constructor
extern "C" DllDecl CTPCC_ODBC* CTPCC_ODBC_new
( LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword,
  LPCSTR szHost, LPCSTR szDatabase );

typedef CTPCC_ODBC* (TYPE_CTPCC_ODBC)(LPCSTR, LPCSTR, LPCSTR,
LPCSTR, LPCSTR);

```

tm_com_dll\src\tpcc_com.h

```

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

#pragma once

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

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

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

public:
    // use this interface for genuine COM errors
    CCOMERR( HRESULT hr )

```

```

{
    m_hr = hr;
    m_iErrorType = 0;
    m_iError = 0;
}

// use this interface to impersonate a non-COM
CCOMERR( int iErrorType, int iError )
{
    m_iErrorType = iErrorType;
    m_iError = iError;
    m_hr = S_OK;
}

int m_hr;
int m_iErrorType;
int m_iError;

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

int ErrorNum() {return m_hr;}

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

};

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

    // COM Interface pointers
    ITPCC*
    m_pNewOrder;

```

```

ITPCC*
m_pPayment; ITPCC*
m_pStockLevel;
ITPCC*
m_pOrderStatus;

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

public:
    CTPCC_COM(BOOL bSinglePool);
    ~CTPCC_COM(void);

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

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

```

```

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

```

```

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

```

```

typedef CTPCC_COM* (TYPE_CTPCC_COM)(BOOL);

```

tm_com_dll\src\tpcc_com.cpp

```

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

```

```

// needed for CoInitializeEx
#define _WIN32_WINNT 0x0400

```

```

#include <windows.h>

```

```

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

```

```

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

```

```

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

```

```

#include "..\..\tpcc_com_all\src\tpcc_com_all_i.c"
// wrapper routine for class constructor
__declspec(dllexport) CTPCC_COM* CTPCC_COM_new(BOOL bSinglePool)
{
    return new CTPCC_COM(bSinglePool);
}

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

    m_bSinglePool = bSinglePool;

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

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

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

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

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

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

        hr = CoCreateInstance(CLSID_Payment, NULL,
            CLSCTX_SERVER, IID_ITPCC, (void **)&m_pPayment);
    }
}

```

```

        if (FAILED(hr)) throw new CCOMERR(hr);

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

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

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

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

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

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

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

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

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

```

```

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

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

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

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

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

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

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

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

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

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

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

```

tpcc_com_ps\src\tpcc_com_ps.def

```

LIBRARY "tpcc_com_ps"

DESCRIPTION 'Proxy/Stub DLL'

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

```

tpcc_com_ps\src\tpcc_com_ps.h

```

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

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


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

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

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

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

#ifndef COM_NO_WINDOWS_H

```

```

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

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

/* Forward Declarations */

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

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

#ifdef __cplusplus
extern "C"{
#endif

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

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

extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

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

EXTERN_C const IID IID_ITPCC;

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

    MIDL_INTERFACE("FEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
    ITPCC : public IUnknown
    {
    public:
        virtual HRESULT __stdcall NewOrder(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

```



```

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

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

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

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

virtual HRESULT __stdcall CallSetComplete( void) = 0;

};

#else          /* C style interface */

typedef struct ITPCCVtbl
{
    BEGIN_INTERFACE

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

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

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

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

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

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

```

```

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

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

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

    END_INTERFACE
} ITPCCVtbl;

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

#ifdef COBJMACROS

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

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

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

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

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

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

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

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

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

```

```

#endif /* COBJMACROS */

#endif          /* C style interface */

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

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

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

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

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

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

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

void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,

```

```

IRpcChannelBuffer *_pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

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

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

HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC __RPC_FAR * This);

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

#endif /* __ITPCC_INTERFACE_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

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

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

```

tpcc_com_ps\src\tpcc_com_ps.idl

```

FILE: ITPCC.IDL
Microsoft TPC-C Kit
Ver. 4.20.000
Copyright Microsoft,
1999
All Rights Reserved
not yet audited

PURPOSE: Defines the interface used by TPCC. This interface
can be implemented by C++ components.

Change history:
4.20.000 - first version

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

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

    HRESULT __stdcall NewOrder

    (

    [in] VARIANT txn_in,

    [out] VARIANT *txn_out

    );

    HRESULT __stdcall Payment

    (

    [in] VARIANT txn_in,

    [out] VARIANT *txn_out

    );

```

```

HRESULT __stdcall Delivery

(

    [in] VARIANT txn_in,

    [out] VARIANT *txn_out

);

HRESULT __stdcall StockLevel

(

    [in] VARIANT txn_in,

    [out] VARIANT *txn_out

);

HRESULT __stdcall OrderStatus

(

    [in] VARIANT txn_in,

    [out] VARIANT *txn_out

);

HRESULT __stdcall CallSetComplete

(

);

// interface ITPCC

```

tpcc_com_ps\src\tpcc_com_ps.i.c

```

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

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

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

```

```

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

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

#ifdef __cplusplus
extern "C"{
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
#endif

```

```

typedef IID CLSID;

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

#endif !_MIDL_USE_GUIDDEF_

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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

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

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

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

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

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

#ifdef __cplusplus
extern "C"{
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

```

```

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

```

tpcc_com_ps\src\tpcc_com_ps_p.c

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

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

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

#if !defined(_M_IA64) && !defined(_M_IX86)
#define USE_STUBLESS_PROXY

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

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

#include "tpcc_com_ps.h"

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

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

typedef struct _MIDL_PROC_FORMAT_STRING
{
    short Pad;
```

```
};

MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */

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

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

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

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

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

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

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

extern const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ];

static const MIDL_STUB_DESC Object_StubDesc =
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.3.280 */
    0,
    UserMarshalRoutines,
```

```

0, /* notify & notify_flag routine table */
0x1, /* MIDL flag */
0, /* Reserved3 */
0, /* Reserved4 */
0 /* Reserved5 */
};

#pragma data_seg(".rdata")

static const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ] =
{
    {
        VARIANT_UserSize
        ,VARIANT_UserMarshal
        ,VARIANT_UserUnmarshal
        ,VARIANT_UserFree
    }

};

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

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

static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
{
    0,
    {

        /* Procedure NewOrder */

0x33, /*
FC_AUTO_HANDLE */
0x6c, /* Old
Flags: object, Oi2 */
/* 2 */ NdrFcLong( 0x0 ), /* 0 */
/* 6 */ NdrFcShort( 0x3 ), /* 3 */
#endif _ALPHA_

```

```

#ifdef _MIPS_
/* 8 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
NdrFcShort( 0x20 ), /* MIPS
Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x20 ), /* PPC
Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x28 ), /* Alpha
Stack size/offset = 40 */
#endif
/* 10 */ NdrFcShort( 0x0 ), /* 0 */
/* 12 */ NdrFcShort( 0x8 ), /* 8 */
/* 14 */ 0x7, /* Oi2 Flags: srv must size, clt must
size, has return, */
0x3, /* 3 */

/* Parameter txn_in */

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

/* Parameter txn_out */

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

```

```

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

/* Return value */

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

/* Procedure Payment */

/* 34 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old
Flags: object, Oi2 */
/* 36 */ NdrFcLong( 0x0 ), /* 0 */
/* 40 */ NdrFcShort( 0x4 ), /* 4 */
#endif _ALPHA_
#endif _PPC_
#if !defined(_MIPS_)
/* 42 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
NdrFcShort( 0x20 ), /* MIPS
Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x20 ), /* PPC
Stack size/offset = 32 */
#endif

```


Stack size/offset = 28 */ #endif #else	NdrFcShort(0x1c), /* PPC	Stack size/offset = 8 */ #endif #else	NdrFcShort(0x8), /* PPC	/* 136 */ 0x33, /* FC_AUTO_HANDLE */ 0x6c, /* Old
Stack size/offset = 32 */ #endif /* 100 */ 0x8, /* FC_LONG */ 0x0, /* 0 */	NdrFcShort(0x20), /* Alpha	Stack size/offset = 8 */ #endif /* 122 */ NdrFcShort(0x3c8), /* Type Offset=968 */ /* Parameter txn_out */	NdrFcShort(0x8), /* Alpha	Flags: object, Oi2 */ /* 138 */ NdrFcLong(0x0), /* 0 */ /* 142 */ NdrFcShort(0x7), /* 7 */ #ifndef _ALPHA_ #ifndef _PPC_ #if !defined(_MIPS_) /* 144 */ NdrFcShort(0x1c), /* x86 Stack size/offset = 28 */ #else
/* Procedure StockLevel */		/* 124 */ NdrFcShort(0x4113), /* Flags: must size, must free, out, simple ref, srv alloc size=16 */ #ifndef _ALPHA_ #ifndef _PPC_ #if !defined(_MIPS_) /* 126 */ NdrFcShort(0x14), /* x86 Stack size/offset = 20 */ #else		Stack size/offset = 32 */ #endif #else
/* 102 */ 0x33, /* FC_AUTO_HANDLE */ 0x6c, /* Old		Stack size/offset = 24 */ #endif #else	NdrFcShort(0x18), /* MIPS	Stack size/offset = 32 */ #endif #else
Flags: object, Oi2 */ /* 104 */ NdrFcLong(0x0), /* 0 */ /* 108 */ NdrFcShort(0x6), /* 6 */ #ifndef _ALPHA_ #ifndef _PPC_ #if !defined(_MIPS_) /* 110 */ NdrFcShort(0x1c), /* x86 Stack size/offset = 28 */ #else	NdrFcShort(0x20), /* MIPS	Stack size/offset = 24 */ #endif #else	NdrFcShort(0x18), /* PPC	Stack size/offset = 32 */ #endif #else
Stack size/offset = 32 */ #endif #else	NdrFcShort(0x20), /* PPC	Stack size/offset = 24 */ #endif /* 128 */ NdrFcShort(0x3da), /* Type Offset=986 */ /* Return value */	NdrFcShort(0x18), /* Alpha	Stack size/offset = 40 */ #endif /* 146 */ NdrFcShort(0x0), /* 0 */ /* 148 */ NdrFcShort(0x8), /* 8 */ /* 150 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has return, */ 0x3, /* 3 */ /* Parameter txn_in */
Stack size/offset = 32 */ #endif #else	NdrFcShort(0x28), /* Alpha	/* 130 */ NdrFcShort(0x70), /* Flags: out, return, base type, */ #ifndef _ALPHA_ #ifndef _PPC_ #if !defined(_MIPS_) /* 132 */ NdrFcShort(0x18), /* x86 Stack size/offset = 24 */ #else		/* 152 */ NdrFcShort(0x8b), /* Flags: must size, must free, in, by val, */ #ifndef _ALPHA_ #ifndef _PPC_ #if !defined(_MIPS_) /* 154 */ NdrFcShort(0x4), /* x86 Stack size/offset = 4 */ #else
Stack size/offset = 40 */ #endif /* 112 */ NdrFcShort(0x0), /* 0 */ /* 114 */ NdrFcShort(0x8), /* 8 */ /* 116 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has return, */ 0x3, /* 3 */		Stack size/offset = 28 */ #endif #else	NdrFcShort(0x1c), /* MIPS	Stack size/offset = 8 */ #endif #else
/* Parameter txn_in */		Stack size/offset = 28 */ #endif #else	NdrFcShort(0x1c), /* PPC	Stack size/offset = 8 */ #endif #else
/* 118 */ NdrFcShort(0x8b), /* Flags: must size, must free, in, by val, */ #ifndef _ALPHA_ #ifndef _PPC_ #if !defined(_MIPS_) /* 120 */ NdrFcShort(0x4), /* x86 Stack size/offset = 4 */ #else	NdrFcShort(0x8), /* MIPS	Stack size/offset = 32 */ #endif /* 134 */ 0x8, /* FC_LONG */ 0x0, /* 0 */	NdrFcShort(0x20), /* Alpha	Stack size/offset = 8 */ #endif /* 156 */ NdrFcShort(0x3c8), /* Type Offset=968 */ /* Parameter txn_out */
Stack size/offset = 8 */ #endif #else		/* Procedure OrderStatus */		/* 158 */ NdrFcShort(0x4113), /* Flags: must size, must free, out, simple ref, srv alloc size=16 */

```

#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 160 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
NdrFcShort( 0x18 ), /* MIPS
Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* PPC
Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* Alpha
Stack size/offset = 24 */
#endif
/* 162 */ NdrFcShort( 0x3da ), /* Type Offset=986 */

/* Return value */

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

/* Procedure CallSetComplete */

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

```

```

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

/* Return value */

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

0x0

}
};

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
0,
{
/* 2 */ NdrFcShort( 0x0 ), /* 0 */
0x12, 0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0x3b0 ), /* Offset= 944 (948) */
/* 6 */ 0x2b, /*
0x9, /*
FC_ULONG */
/* 8 */ 0x7, /* Corr desc: FC_USHORT */
0x0, /* */
/* 10 */ NdrFcShort( 0xffff8 ), /* -8 */
/* 12 */ NdrFcShort( 0x2 ), /* Offset= 2 (14) */
/* 14 */ NdrFcShort( 0x10 ), /* 16 */
/* 16 */ NdrFcShort( 0x2b ), /* 43 */
/* 18 */ NdrFcLong( 0x3 ), /* 3 */
/* 22 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 24 */ NdrFcLong( 0x11 ), /* 17 */
/* 28 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE */
/* 30 */ NdrFcLong( 0x2 ), /* 2 */
/* 34 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 36 */ NdrFcLong( 0x4 ), /* 4 */
/* 40 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT */
/* 42 */ NdrFcLong( 0x5 ), /* 5 */
/* 46 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */

```

```

/* 48 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 54 */ NdrFcLong( 0xa ), /* 10 */
/* 58 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 60 */ NdrFcLong( 0x6 ), /* 6 */
/* 64 */ NdrFcShort( 0xd6 ), /* Offset= 214 (278) */
/* 66 */ NdrFcLong( 0x7 ), /* 7 */
/* 70 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 72 */ NdrFcLong( 0x8 ), /* 8 */
/* 76 */ NdrFcShort( 0xd0 ), /* Offset= 208 (284) */
/* 78 */ NdrFcLong( 0xd ), /* 13 */
/* 82 */ NdrFcShort( 0xe2 ), /* Offset= 226 (308) */
/* 84 */ NdrFcLong( 0x9 ), /* 9 */
/* 88 */ NdrFcShort( 0xee ), /* Offset= 238 (326) */
/* 90 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 94 */ NdrFcShort( 0xfa ), /* Offset= 250 (344) */
/* 96 */ NdrFcLong( 0x24 ), /* 36 */
/* 100 */ NdrFcShort( 0x308 ), /* Offset= 776 (876) */
/* 102 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 106 */ NdrFcShort( 0x302 ), /* Offset= 770 (876) */
/* 108 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 112 */ NdrFcShort( 0x300 ), /* Offset= 768 (880) */
/* 114 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 118 */ NdrFcShort( 0x2fe ), /* Offset= 766 (884) */
/* 120 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 124 */ NdrFcShort( 0x2fc ), /* Offset= 764 (888) */
/* 126 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 130 */ NdrFcShort( 0x2fa ), /* Offset= 762 (892) */
/* 132 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 136 */ NdrFcShort( 0x2f8 ), /* Offset= 760 (896) */
/* 138 */ NdrFcLong( 0x400b ), /* 16395 */
/* 142 */ NdrFcShort( 0x2e6 ), /* Offset= 742 (884) */
/* 144 */ NdrFcLong( 0x400a ), /* 16394 */
/* 148 */ NdrFcShort( 0x2e4 ), /* Offset= 740 (888) */
/* 150 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 154 */ NdrFcShort( 0x2ea ), /* Offset= 746 (900) */
/* 156 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 160 */ NdrFcShort( 0x2e0 ), /* Offset= 736 (896) */
/* 162 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 166 */ NdrFcShort( 0x2e2 ), /* Offset= 738 (904) */
/* 168 */ NdrFcLong( 0x400d ), /* 16397 */
/* 172 */ NdrFcShort( 0x2e0 ), /* Offset= 736 (908) */
/* 174 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 178 */ NdrFcShort( 0x2de ), /* Offset= 734 (912) */
/* 180 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 184 */ NdrFcShort( 0x2dc ), /* Offset= 732 (916) */
/* 186 */ NdrFcLong( 0x400c ), /* 16396 */
/* 190 */ NdrFcShort( 0x2da ), /* Offset= 730 (920) */
/* 192 */ NdrFcLong( 0x10 ), /* 16 */
/* 196 */ NdrFcShort( 0x8002 ), /* Simple arm type: FC_CHAR */
/* 198 */ NdrFcLong( 0x12 ), /* 18 */
/* 202 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 204 */ NdrFcLong( 0x13 ), /* 19 */
/* 208 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */

```



```

/* 210 */ NdrFcLong( 0x16 ), /* 22 */
/* 214 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 216 */ NdrFcLong( 0x17 ), /* 23 */
/* 220 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 222 */ NdrFcLong( 0xe ), /* 14 */
/* 226 */ NdrFcShort( 0x2be ), /* Offset= 702 (928) */
/* 228 */ NdrFcLong( 0x400e ), /* 16398 */
/* 232 */ NdrFcShort( 0x2c4 ), /* Offset= 708 (940) */
/* 234 */ NdrFcLong( 0x4010 ), /* 16400 */
/* 238 */ NdrFcShort( 0x2c2 ), /* Offset= 706 (944) */
/* 240 */ NdrFcLong( 0x4012 ), /* 16402 */
/* 244 */ NdrFcShort( 0x280 ), /* Offset= 640 (884) */
/* 246 */ NdrFcLong( 0x4013 ), /* 16403 */
/* 250 */ NdrFcShort( 0x27e ), /* Offset= 638 (888) */
/* 252 */ NdrFcLong( 0x4016 ), /* 16406 */
/* 256 */ NdrFcShort( 0x278 ), /* Offset= 632 (888) */
/* 258 */ NdrFcLong( 0x4017 ), /* 16407 */
/* 262 */ NdrFcShort( 0x272 ), /* Offset= 626 (888) */
/* 264 */ NdrFcLong( 0x0 ), /* 0 */
/* 268 */ NdrFcShort( 0x0 ), /* Offset= 0 (268) */
/* 270 */ NdrFcLong( 0x1 ), /* 1 */
/* 274 */ NdrFcShort( 0x0 ), /* Offset= 0 (274) */
/* 276 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (275) */
/* 278 */

FC_STRUCT */
/* 280 */ NdrFcShort( 0x8 ), /* 8 */
/* 282 */ 0xb, /* FC_HYPER */
/* 284 */

FC_END */
/* 286 */ NdrFcShort( 0xc ), /* 12 */
/* 288 */

FC_CARRY */
/* 290 */ NdrFcShort( 0x2 ), /* 2 */
/* 292 */ 0x9, /* Corr desc: FC_ULONG */
/* 294 */ NdrFcShort( 0xffc ), /* -4 */
/* 296 */ 0x6, /* FC_SHORT */
/* 298 */

FC_END */
/* 300 */ NdrFcShort( 0x8 ), /* 8 */
/* 302 */ NdrFcShort( 0xffffffff2 ), /* Offset= -14 (288) */
/* 304 */ 0x8, /* FC_LONG */

```

```

FC_CONSTANT */ 0x5c,
FC_END */
/* 308 */

*/
FC_CONSTANT_IID */
/* 310 */ NdrFcLong( 0x0 ), /* 0 */
/* 314 */ NdrFcShort( 0x0 ), /* 0 */
/* 316 */ NdrFcShort( 0x0 ), /* 0 */
/* 318 */ 0xc0, /* 192 */
/* 320 */ 0x0, /* 0 */
/* 322 */ 0x0, /* 0 */
/* 324 */ 0x0, /* 0 */
/* 326 */

*/
FC_CONSTANT_IID */
/* 328 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 332 */ NdrFcShort( 0x0 ), /* 0 */
/* 334 */ NdrFcShort( 0x0 ), /* 0 */
/* 336 */ 0xc0, /* 192 */
/* 338 */ 0x0, /* 0 */
/* 340 */ 0x0, /* 0 */
/* 342 */ 0x0, /* 0 */
/* 344 */

[pointer_deref] */
/* 346 */ NdrFcShort( 0x2 ), /* Offset= 2 (348) */
/* 348 */

/* 350 */ NdrFcShort( 0x1fc ), /* 0x12, 0x0, /* FC_UP */
/* 352 */ /* Offset= 508 (858) */

FC_ENCAPSULATED_UNION */
/* 354 */ NdrFcShort( 0x18 ), /* 24 */
/* 356 */ NdrFcShort( 0xa ), /* 10 */
/* 358 */ NdrFcLong( 0x8 ), /* 8 */
/* 362 */ NdrFcShort( 0x58 ), /* Offset= 88 (450) */
/* 364 */ NdrFcLong( 0xd ), /* 13 */
/* 368 */ NdrFcShort( 0x78 ), /* Offset= 120 (488) */

```

```

/* 370 */ NdrFcShort( 0x9 ), /* Offset= 148 (522) */
/* 376 */ NdrFcLong( 0xc ), /* 12 */
/* 380 */ NdrFcShort( 0xbc ), /* Offset= 188 (568) */
/* 382 */ NdrFcLong( 0x24 ), /* 36 */
/* 386 */ NdrFcShort( 0x114 ), /* Offset= 276 (662) */
/* 388 */ NdrFcLong( 0x800d ), /* 32781 */
/* 392 */ NdrFcShort( 0x130 ), /* Offset= 304 (696) */
/* 394 */ NdrFcLong( 0x10 ), /* 16 */
/* 398 */ NdrFcShort( 0x148 ), /* Offset= 328 (726) */
/* 400 */ NdrFcLong( 0x2 ), /* 2 */
/* 404 */ NdrFcShort( 0x160 ), /* Offset= 352 (756) */
/* 406 */ NdrFcLong( 0x3 ), /* 3 */
/* 410 */ NdrFcShort( 0x178 ), /* Offset= 376 (786) */
/* 412 */ NdrFcLong( 0x14 ), /* 20 */
/* 416 */ NdrFcShort( 0x190 ), /* Offset= 400 (816) */
/* 418 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (417) */
/* 420 */

0x1b, /*
FC_CARRY */
/* 422 */ NdrFcShort( 0x4 ), /* 4 */
/* 424 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */
/* 426 */ NdrFcShort( 0x0 ), /* 0 */
/* 428 */

0x4b, /* FC_PP
0x5c, /*
FC_PAD */
/* 430 */

0x48, /*
FC_VARIABLE_REPEAT */
/* 432 */ NdrFcShort( 0x4 ), /* 4 */
/* 434 */ NdrFcShort( 0x0 ), /* 0 */
/* 436 */ NdrFcShort( 0x1 ), /* 1 */
/* 438 */ NdrFcShort( 0x0 ), /* 0 */
/* 440 */ NdrFcShort( 0x0 ), /* 0 */
/* 442 */ 0x12, 0x0, /* FC_UP */
/* 444 */ NdrFcShort( 0xffffffff6e ), /* Offset= -146 (298) */
/* 446 */

0x5b, /*
FC_END */

0x8, /*
FC_LONG */
/* 448 */ 0x5c, /* FC_PAD */
/* 450 */

0x5b, /*
FC_END */
/* 450 */

```

FC_PSTRUCT */	0x16,	/*	FC_POINTER */	0x36,	/*	/* 548 */	0x48,	/*
/* 452 */ NdrFcShort(0x8),	0x3,	/* 3 */	FC_END */	/* FC_PAD */	/*	FC_VARIABLE_REPEAT */	0x49,	/*
/* 454 */			/* 500 */	0x5b,	/*			
/	0x4b,	/ FC_PP	/* 502 */ NdrFcShort(0xffffffff),	0x11, 0x0, /* FC_RP */		FC_FIXED_OFFSET */		
	0x5c,	/*	/* 504 */	/* Offset= -32 (470) */		/* 550 */ NdrFcShort(0x4),	/* 4 */	
FC_PAD */						/* 552 */ NdrFcShort(0x0),	/* 0 */	
/* 456 */			FC_BOGUS_ARRAY */	0x21,	/*	/* 554 */ NdrFcShort(0x1),	/* 1 */	
						/* 556 */ NdrFcShort(0x0),	/* 0 */	
FC_NO_REPEAT */	0x46,	/*	/* 506 */ NdrFcShort(0x0),	0x3,	/* 3 */	/* 558 */ NdrFcShort(0x0),	/* 0 */	
	0x5c,	/*	/* 508 */ 0x19,	/* 0 */		/* 560 */ 0x12, 0x0, /* FC_UP */		
FC_PAD */			FC_ULONG */	/* Corr desc: field pointer,		/* 562 */ NdrFcShort(0x182),	/* Offset= 386 (948) */	
/* 458 */ NdrFcShort(0x4),	/* 4 */					/* 564 */	0x5b,	/*
/* 460 */ NdrFcShort(0x4),	/* 4 */		/* 510 */ NdrFcShort(0x0),	0x0,	/* */	FC_END */		
/* 462 */ 0x11, 0x0, /* FC_RP */			/* 512 */ NdrFcLong(0xffffffff),	/* 0 */			0x8,	/*
/* 464 */ NdrFcShort(0xffffffff),	/* Offset= -44 (420) */		/* 516 */ 0x4c,	/* -1 */		FC_LONG */		
/* 466 */				/* FC_EMBEDDED_COMPLEX */		/* 566 */ 0x5c,	/* FC_PAD */	
	0x5b,	/*	/* 518 */ NdrFcShort(0xffffffff),	0x0,	/* 0 */		0x5b,	/*
FC_END */			/* 520 */ 0x5c,	/* Offset= -192 (326) */		FC_END */		
				/* FC_PAD */		/* 568 */	0x1a,	/*
	0x8,	/*	FC_END */	0x5b,	/*	FC_BOGUS_STRUCT */		
/* 468 */ 0x8,	/* FC_LONG */		/* 522 */				0x3,	/* 3 */
	0x5b,	/*	FC_BOGUS_STRUCT */	0x1a,	/*	/* 570 */ NdrFcShort(0x8),	/* 8 */	
FC_END */						/* 572 */ NdrFcShort(0x0),	/* 0 */	
/* 470 */			/* 524 */ NdrFcShort(0x8),	0x3,	/* 3 */	/* 574 */ NdrFcShort(0x6),	/* Offset= 6 (580) */	
	0x21,	/*	/* 526 */ NdrFcShort(0x0),	/* 8 */		/* 576 */ 0x8,	/* FC_LONG */	
FC_BOGUS_ARRAY */			/* 528 */ NdrFcShort(0x6),	/* 0 */			0x36,	/*
	0x3,	/* 3 */	/* 530 */ 0x8,	/* Offset= 6 (534) */		FC_POINTER */		
/* 472 */ NdrFcShort(0x0),	/* 0 */			/* FC_LONG */		/* 578 */ 0x5c,	/* FC_PAD */	
/* 474 */ 0x19,	/* Corr desc: field pointer,		FC_POINTER */	0x36,	/*		0x5b,	/*
FC_ULONG */			/* 532 */ 0x5c,	/* FC_PAD */		FC_END */		
	0x0,	/* */		0x5b,	/*	/* 580 */		
/* 476 */ NdrFcShort(0x0),	/* 0 */		FC_END */			/* 582 */ NdrFcShort(0xffffffff),	0x11, 0x0, /* FC_RP */	
/* 478 */ NdrFcLong(0xffffffff),	/* -1 */		/* 534 */			/* 584 */	/* Offset= -44 (538) */	
/* 482 */ 0x4c,	/* FC_EMBEDDED_COMPLEX */						0x2f,	/* FC_IP
	0x0,	/* 0 */	/* 536 */ NdrFcShort(0xffffffff),	0x11, 0x0, /* FC_RP */			0x5a,	/*
/* 484 */ NdrFcShort(0xffffffff),	/* Offset= -176 (308) */		/* 538 */	/* Offset= -32 (504) */		FC_CONSTANT_IID */		
/* 486 */ 0x5c,	/* FC_PAD */					/* 586 */ NdrFcLong(0x2f),	/* 47 */	
	0x5b,	/*	FC_CARRAY */	0x1b,	/*	/* 590 */ NdrFcShort(0x0),	/* 0 */	
FC_END */						/* 592 */ NdrFcShort(0x0),	/* 0 */	
/* 488 */			/* 540 */ NdrFcShort(0x4),	0x3,	/* 3 */	/* 594 */ 0xc0,	/* 192 */	
	0x1a,	/*	/* 542 */ 0x19,	/* 4 */			0x0,	/* 0 */
FC_BOGUS_STRUCT */			FC_ULONG */	/* Corr desc: field pointer,		/* 596 */ 0x0,	/* 0 */	
	0x3,	/* 3 */				/* 598 */ 0x0,	/* 0 */	
/* 490 */ NdrFcShort(0x8),	/* 8 */		/* 544 */ NdrFcShort(0x0),	0x0,	/* */	/* 600 */ 0x0,	/* 0 */	
/* 492 */ NdrFcShort(0x0),	/* 0 */		/* 546 */	/* 0 */			0x0,	/* 0 */
/* 494 */ NdrFcShort(0x6),	/* Offset= 6 (500) */						0x0,	/* 0 */
/* 496 */ 0x8,	/* FC_LONG */		*/	0x4b,	/* FC_PP		/* 0 */	
				0x5c,	/*	/* 602 */	0x46,	/* 70 */
			FC_PAD */					

FC_CARRAY */	0x1b,	/*	/* 650 */ NdrFcShort(0x0),	/* 0 */	FC_BOGUS_STRUCT */	0x1a,	/*
/* 604 */ NdrFcShort(0x1),	0x0,	/* 0 */	/* 654 */ 0x12, 0x0, /* FC_UP */		/* 698 */ NdrFcShort(0x18),	0x3,	/* 3 */
/* 606 */ 0x19,	/* 1 */		/* 656 */ NdrFcShort(0xfffffd4),	/* Offset= -44 (612) */	/* 700 */ NdrFcShort(0x0),	/* 24 */	
FC_ULONG */	/* Corr desc: field pointer,		/* 658 */		/* 702 */ NdrFcShort(0xa),	/* 0 */	
	0x0,	/* */	FC_END */	0x5b,	/* 704 */ 0x8,	/* Offset= 10 (712) */	
/* 608 */ NdrFcShort(0x4),	/* 4 */					/* FC_LONG */	/*
/* 610 */ 0x1,	/* FC_BYTE */		FC_LONG */	0x8,	FC_POINTER */	0x36,	/*
	0x5b,	/*	/* 660 */ 0x5c,	/* FC_PAD */	/* 706 */ 0x4c,	/* FC_EMBEDDED_COMPLEX */	
FC_END */			FC_END */	0x5b,	/* 708 */ NdrFcShort(0xfffffe8),	0x0,	/* 0 */
/* 612 */	0x1a,	/*	/* 662 */		/* 710 */ 0x5c,	/* Offset= -24 (684) */	
FC_BOGUS_STRUCT */			FC_BOGUS_STRUCT */	0x1a,		/* FC_PAD */	/*
/* 614 */ NdrFcShort(0x10),	/* 3 */				FC_END */	0x5b,	/*
/* 616 */ NdrFcShort(0x0),	/* 16 */		/* 664 */ NdrFcShort(0x8),	/* 3 */	/* 712 */		
/* 618 */ NdrFcShort(0xa),	/* 0 */		/* 666 */ NdrFcShort(0x0),	/* 8 */	/* 714 */ NdrFcShort(0xfffff0c),	0x11, 0x0,	/* FC_RP */
/* 620 */ 0x8,	/* Offset= 10 (628) */		/* 668 */ NdrFcShort(0x6),	/* 0 */	/* 716 */	/* Offset= -244 (470) */	
	/* FC_LONG */		/* 670 */ 0x8,	/* Offset= 6 (674) */			
FC_LONG */	0x8,	/*	/* FC_LONG */	/* FC_LONG */	FC_CARRAY */	0x1b,	/*
/* 622 */ 0x4c,	/* FC_EMBEDDED_COMPLEX */		FC_POINTER */	0x36,	/* 718 */ NdrFcShort(0x1),	/* 0 */	
/* 624 */ NdrFcShort(0xfffffd8),	/* 0 */		/* 672 */ 0x5c,	/* FC_PAD */	/* 720 */ 0x19,	/* 1 */	
/* 626 */ 0x36,	/* Offset= -40 (584) */		FC_END */	0x5b,	FC_ULONG */	/* Corr desc: field pointer,	
	/* FC_POINTER */		/* 674 */				
FC_END */	0x5b,	/*	/* 676 */ NdrFcShort(0xfffffd4),	/* Offset= -44 (632) */	/* 722 */ NdrFcShort(0x0),	0x0,	/* */
/* 628 */			/* 678 */		/* 724 */ 0x1,	/* 0 */	
/* 630 */ NdrFcShort(0xfffffe4),	0x12, 0x0, /* FC_UP */		FC_SMFARRAY */	0x1d,	FC_END */	/* FC_BYTE */	/*
/* 632 */	/* Offset= -28 (602) */				/* 726 */	0x5b,	/*
	0x1b,	/*	/* 680 */ NdrFcShort(0x8),	/* 0 */			
FC_CARRAY */	0x3,	/* 3 */	/* 682 */ 0x2,	/* 8 */	FC_PSTRUCT */	0x16,	/*
/* 634 */ NdrFcShort(0x4),	/* 4 */			/* FC_CHAR */			
/* 636 */ 0x19,	/* Corr desc: field pointer,		FC_END */	0x5b,	/* 728 */ NdrFcShort(0x8),	0x3,	/* 3 */
FC_ULONG */			/* 684 */		/* 730 */	/* 8 */	
	0x0,	/* */	FC_STRUCT */	0x15,	*/	0x4b,	/* FC_PP
/* 638 */ NdrFcShort(0x0),	/* 0 */					0x5c,	/*
/* 640 */	0x4b,	/* FC_PP	/* 686 */ NdrFcShort(0x10),	/* 16 */	FC_PAD */		
/	0x5c,	/	/* 688 */ 0x8,	/* FC_LONG */	/* 732 */		
				0x6,			
FC_PAD */			FC_SHORT */	/* FC_SHORT */	FC_NO_REPEAT */	0x5c,	/*
/* 642 */	0x48,	/*	/* 690 */ 0x6,	0x4c,			
FC_VARIABLE_REPEAT */	0x49,	/*	FC_EMBEDDED_COMPLEX */	/* 0 */	FC_PAD */	0x46,	/*
			/* 692 */ 0x0,	NdrFcShort(0xfffffff1),	/* 734 */ NdrFcShort(0x4),		
FC_FIXED_OFFSET */			Offset= -15 (678) */		/* 736 */ NdrFcShort(0x4),	/* 4 */	
/* 644 */ NdrFcShort(0x4),	/* 4 */		FC_END */	0x5b,	/* 738 */ 0x12, 0x0, /* FC_UP */	/* 4 */	
/* 646 */ NdrFcShort(0x0),	/* 0 */		/* 696 */		/* 740 */ NdrFcShort(0xffffffe8),	/* Offset= -24 (716) */	
/* 648 */ NdrFcShort(0x1),	/* 1 */				/* 742 */		
					FC_END */	0x5b,	/*

FC_LONG */	0x8,	/*	/* 782 */ NdrFcShort(0x0),	0x0, /*	/* */	FC_END */	0x5c,	/*
/* 744 */ 0x8,	/* FC_LONG */		/* 784 */ 0x8,	/* FC_LONG */		/* 824 */	0x46,	/*
FC_END */	0x5b,	/*	FC_END */	0x5b,	/*	FC_NO_REPEAT */	0x5c,	/*
/* 746 */			/* 786 */			FC_PAD */		
FC_CARRY */	0x1b,	/*	FC_PSTRUCT */	0x16,	/*	/* 824 */ NdrFcShort(0x4),	/* 4 */	
						/* 826 */ NdrFcShort(0x4),	/* 4 */	
/* 748 */ NdrFcShort(0x2),	0x1,	/* 1 */	/* 788 */ NdrFcShort(0x8),	0x3,	/* 3 */	/* 828 */ 0x12, 0x0, /* FC_UP */		
/* 750 */ 0x19,	/* 2 */		/* 790 */	/* 8 */		/* 830 */ NdrFcShort(0xffffffff),	/* Offset= -24 (806) */	
FC_ULONG */	/* Corr desc: field pointer,		*/	0x4b,	/* FC_PP	/* 832 */		
	0x0,	/* */	FC_PAD */	0x5c,	/*	FC_END */	0x5b,	/*
/* 752 */ NdrFcShort(0x0),	/* 0 */		/* 792 */					
/* 754 */ 0x6,	/* FC_SHORT */			0x46,	/*	FC_LONG */	0x8,	/*
	0x5b,	/*	FC_NO_REPEAT */			/* 834 */ 0x8,	/* FC_LONG */	/*
FC_END */				0x5c,	/*	FC_END */	0x5b,	/*
/* 756 */			FC_PAD */			/* 836 */		
FC_PSTRUCT */	0x16,	/*	/* 794 */ NdrFcShort(0x4),	/* 4 */				
			/* 796 */ NdrFcShort(0x4),	/* 4 */		FC_STRUCT */	0x15,	/*
/* 758 */ NdrFcShort(0x8),	/* 8 */	/* 3 */	/* 798 */ 0x12, 0x0, /* FC_UP */					
/* 760 */			/* 800 */ NdrFcShort(0xffffffff),	/* Offset= -24 (776) */		/* 838 */ NdrFcShort(0x8),	0x3,	/* 3 */
	0x4b,	/* FC_PP	/* 802 */			/* 840 */ 0x8,	/* 8 */	
/				0x5b,	/	/* FC_LONG */	0x8,	/*
FC_PAD */	0x5c,	/*	FC_END */			FC_LONG */		
/* 762 */				0x8,	/*	/* 842 */ 0x5c,	/* FC_PAD */	
FC_NO_REPEAT */	0x46,	/*	FC_LONG */			FC_END */	0x5b,	/*
	0x5c,	/*	/* 804 */ 0x8,	/* FC_LONG */		/* 844 */		
FC_PAD */			FC_END */				0x1b,	/*
/* 764 */ NdrFcShort(0x4),	/* 4 */		/* 806 */			FC_CARRY */		
/* 766 */ NdrFcShort(0x4),	/* 4 */			0x1b,	/*	/* 846 */ NdrFcShort(0x8),	0x3,	/* 3 */
/* 768 */ 0x12, 0x0, /* FC_UP */			FC_CARRY */			/* 848 */ 0x7,	/* 8 */	
/* 770 */ NdrFcShort(0xffffffff),	/* Offset= -24 (746) */		/* 808 */ NdrFcShort(0x8),	/* 8 */	/* 7 */	/* Corr desc: FC_USHORT */	0x0,	/* */
/* 772 */			/* 810 */ 0x19,	/* Corr desc: field pointer,		/* 850 */ NdrFcShort(0xffd8),	/* -40 */	
FC_END */	0x5b,	/*	FC_ULONG */			/* 852 */ 0x4c,	/* FC_EMBEDDED_COMPLEX */	
				0x0,	/* */	/* 854 */ NdrFcShort(0xffffffff),	/* Offset= -18 (836) */	
FC_LONG */	0x8,	/*	/* 812 */ NdrFcShort(0x0),	/* 0 */		/* 856 */ 0x5c,	/* FC_PAD */	/*
/* 774 */ 0x8,	/* FC_LONG */		/* 814 */ 0xb,	/* FC_HYPER */		FC_END */	0x5b,	/*
FC_END */	0x5b,	/*	FC_END */			/* 858 */		
/* 776 */			/* 816 */	0x16,	/*	FC_BOGUS_STRUCT */	0x1a,	/*
FC_CARRY */	0x1b,	/*	FC_PSTRUCT */					
				0x3,	/* 3 */	/* 860 */ NdrFcShort(0x28),	0x3,	/* 3 */
/* 778 */ NdrFcShort(0x4),	/* 4 */	/* 3 */	/* 818 */ NdrFcShort(0x8),	/* 8 */		/* 862 */ NdrFcShort(0xffffffff),	/* 40 */	
/* 780 */ 0x19,	/* Corr desc: field pointer,		/* 820 */			/* Offset= -18 (844) */	/* Offset= 0 (864) */	
FC_ULONG */			*/	0x4b,	/* FC_PP	/* 864 */ NdrFcShort(0x0),	/* FC_SHORT */	
						/* 866 */ 0x6,		

FC_SHORT */	0x6,	/*	/* 908 */	NdrFcShort(0xfffffd92),	/* Offset= -622 (284) */	FC_SHORT */	0x8,	/*
/* 868 */	0x38,	/* FC_ALIGNM4 */	[pointer_deref] */	0x12, 0x10,	/* FC_UP	/* 960 */	0x6,	/*
FC_LONG */	0x8,	/* FC_LONG */	/* 910 */	NdrFcShort(0xfffffda6),	/* Offset= -602 (308) */	FC_SHORT */	0x6,	/*
/* 870 */	0x4c,	/* FC_LONG */	/* 912 */	0x12, 0x10,	/* FC_UP	/* 962 */	0x4c,	/* FC_EMBEDDED_COMPLEX */
FC_EMBEDDED_COMPLEX */	0x0,	/* 0 */	[pointer_deref] */	/* 914 */	NdrFcShort(0xfffffdb4),	/* Offset= -588 (326) */	0x0,	/* 0 */
/* 872 */	NdrFcShort(0xfffffdf7),	/*	/* 916 */	0x12, 0x10,	/* FC_UP	/* 964 */	NdrFcShort(0xfffffc42),	/* Offset= -958 (6) */
Offset= -521 (352) */	0x5b,	/*	[pointer_deref] */	/* 918 */	NdrFcShort(0xfffffdc2),	/* Offset= -574 (344) */	/* 966 */	0x5c,
FC_END */	0x12, 0x0,	/* FC_UP */	/* 920 */	0x12, 0x10,	/* FC_UP	FC_END */	/* 968 */	0xb4,
/* 876 */	/* Offset= -266 (612) */	/*	[pointer_deref] */	/* 922 */	NdrFcShort(0x2),	/* Offset= 2 (924) */	/* 970 */	NdrFcShort(0x0),
/* 878 */	0x12, 0x8,	/* FC_UP	/* 924 */	0x12, 0x0,	/* FC_UP */	/* 972 */	NdrFcShort(0x10),	/* 16 */
[simple_pointer] */	/* FC_BYTE */	/*	/* 926 */	NdrFcShort(0x16),	/* Offset= 22 (948) */	/* 974 */	NdrFcShort(0x0),	/* 0 */
/* 882 */	0x5c,	/*	/* 928 */	0x15,	/*	/* 976 */	NdrFcShort(0xfffffc32),	/* Offset= -974 (2) */
FC_PAD */	0x12, 0x8,	/* FC_UP	FC_STRUCT */	0x7,	/* 7 */	/* 978 */	0x11, 0x4,	/* FC_RP
/* 884 */	/* FC_SHORT */	/*	/* 930 */	NdrFcShort(0x10),	/* 16 */	[allocated_on_stack] */	/* 980 */	NdrFcShort(0x6),
[simple_pointer] */	0x5c,	/*	/* 932 */	0x6,	/* FC_SHORT */	/* 982 */	/* Offset= 6 (986) */	0x83,
/* 886 */	0x12, 0x8,	/* FC_UP	FC_BYTE */	/* 934 */	0x1,	/* FC_BYTE */	/* 984 */	NdrFcShort(0xfffffdc),
FC_PAD */	0x12, 0x8,	/* FC_UP	/* 936 */	0x8,	/* FC_LONG */	/* 986 */	0xb4,	/* FC_USER_MARSHAL */
/* 888 */	/* FC_LONG */	/*	FC_ALIGNM8 */	/* 938 */	0xb,	/* FC_HYPER */	0x83,	/* 131 */
[simple_pointer] */	0x5c,	/*	/* 940 */	FC_END */	0x5b,	/*	/* 988 */	NdrFcShort(0x0),
/* 890 */	0x12, 0x8,	/* FC_UP	/* 942 */	NdrFcShort(0xfffffff2),	/* Offset= -14 (928) */	/* 990 */	NdrFcShort(0x10),	/* 16 */
FC_PAD */	0x12, 0x8,	/* FC_UP	/* 944 */	0x12, 0x8,	/* FC_UP	/* 992 */	NdrFcShort(0x0),	/* 0 */
/* 894 */	/* FC_FLOAT */	/*	[simple_pointer] */	/* 946 */	0x2,	/* FC_CHAR */	/* 994 */	NdrFcShort(0xfffffff4),
FC_PAD */	0x5c,	/*	/* 948 */	FC_PAD */	0x5c,	/*	0x0	};
/* 896 */	0x12, 0x8,	/* FC_UP	FC_BOGUS_STRUCT */	0x1a,	/*	const CInterfaceProxyVtbl * _tpcc_com_ps_ProxyVtblList[] =	{	(CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
[simple_pointer] */	0x12, 0x0,	/* FC_UP */	/* 950 */	NdrFcShort(0x20),	/* 32 */	0	};	const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[] =
/* 898 */	/* FC_DOUBLE */	/*	/* 952 */	NdrFcShort(0x0),	/* 0 */	0	};	{
FC_PAD */	0x12, 0x10,	/* FC_UP	/* 954 */	NdrFcShort(0x0),	/* Offset= 0 (954) */	PCInterfaceName const _tpcc_com_ps_InterfaceNamesList[] =	{	"ITPCC",
/* 900 */	0x12, 0x10,	/* FC_UP	/* 956 */	0x8,	/* FC_LONG */	0	}	
[pointer_deref] */								

```

};

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

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{
    if(!_tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }

    return 0;
}

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

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

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

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

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

```

```

#@MIDL_FILE_HEADING( )

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

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

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

#include "tpcc_com_ps.h"

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

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

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

extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */

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

```

```

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

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

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

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

CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    (void *)-1 /* ITPCC::NewOrder */ ,
    (void *)-1 /* ITPCC::Payment */ ,

```



```

#ifndef _ALPHA_
/* 72 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
NdrFcShort( 0x8 ), /* xpp64

Stack size/offset = 8 */
#endif
/* 74 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

/* Parameter txn_out */

/* 76 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out,
simple ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 78 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
NdrFcShort( 0x20 ), /* xpp64

Stack size/offset = 32 */
#endif
/* 80 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Return value */

/* 82 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
/* 84 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
NdrFcShort( 0x28 ), /* xpp64

Stack size/offset = 40 */
#endif
/* 86 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure Delivery */

/* 88 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old

Flags: object, Oi2 */
/* 90 */ NdrFcLong( 0x0 ), /* 0 */
/* 94 */ NdrFcShort( 0x5 ), /* 5 */
#ifndef _ALPHA_
/* 96 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
NdrFcShort( 0x30 ), /* xpp64

Stack size/offset = 48 */
#endif
/* 98 */ NdrFcShort( 0x0 ), /* 0 */
/* 100 */ NdrFcShort( 0x8 ), /* 8 */
/* 102 */ 0x47, /* Oi2 Flags: srv must size, clt must
size, has return, has ext, */
0x3, /* 3 */
/* 104 */ 0xa, /* 10 */

```

```

/* 96 */ NdrFcShort( 0x20 ), /* 0x7, /* Ext
Flags: new corr desc, clt corr check, srv corr check, */
/* 108 */ NdrFcShort( 0x20 ), /* 32 */
/* 110 */ NdrFcShort( 0x0 ), /* 0 */
/* 112 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 114 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in,
by val, */
#ifndef _ALPHA_
/* 116 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
NdrFcShort( 0x8 ), /* xpp64

Stack size/offset = 8 */
#endif
/* 118 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

/* Parameter txn_out */

/* 120 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out,
simple ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 122 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
NdrFcShort( 0x20 ), /* xpp64

Stack size/offset = 32 */
#endif
/* 124 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Return value */

/* 126 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
/* 128 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
NdrFcShort( 0x28 ), /* xpp64

Stack size/offset = 40 */
#endif
/* 130 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure StockLevel */

/* 132 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old

Flags: object, Oi2 */
/* 134 */ NdrFcLong( 0x0 ), /* 0 */
/* 138 */ NdrFcShort( 0x6 ), /* 6 */
#ifndef _ALPHA_
/* 140 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else

```

```

Stack size/offset = 48 */
/* 142 */ NdrFcShort( 0x0 ), /* 0 */
/* 144 */ NdrFcShort( 0x8 ), /* 8 */
/* 146 */ 0x47, /* Oi2 Flags: srv must size, clt must
size, has return, has ext, */
0x3, /* 3 */
/* 148 */ 0xa, /* 10 */
0x7, /* Ext

Flags: new corr desc, clt corr check, srv corr check, */
/* 150 */ NdrFcShort( 0x20 ), /* 32 */
/* 152 */ NdrFcShort( 0x20 ), /* 32 */
/* 154 */ NdrFcShort( 0x0 ), /* 0 */
/* 156 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 158 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in,
by val, */
#ifndef _ALPHA_
/* 160 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
NdrFcShort( 0x8 ), /* xpp64

Stack size/offset = 8 */
#endif
/* 162 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

/* Parameter txn_out */

/* 164 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out,
simple ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 166 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
NdrFcShort( 0x20 ), /* xpp64

Stack size/offset = 32 */
#endif
/* 168 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Return value */

/* 170 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
/* 172 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
NdrFcShort( 0x28 ), /* xpp64

Stack size/offset = 40 */
#endif
/* 174 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure OrderStatus */

```



```

/* 176 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old
Flags: object, Oi2 */
/* 178 */ NdrFcLong( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x7 ), /* 7 */
#ifdef _ALPHA_
/* 184 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
NdrFcShort( 0x30 ), /* xpp64
Stack size/offset = 48 */
#endif
/* 186 */ NdrFcShort( 0x0 ), /* 0 */
/* 188 */ NdrFcShort( 0x8 ), /* 8 */
/* 190 */ 0x47, /* Oi2 Flags: srv must size, clt must
size, has return, has ext, */
0x3, /* 3 */
/* 192 */ 0xa, /* 10 */
0x7, /* Ext
Flags: new corr desc, clt corr check, srv corr check, */
/* 194 */ NdrFcShort( 0x20 ), /* 32 */
/* 196 */ NdrFcShort( 0x20 ), /* 32 */
/* 198 */ NdrFcShort( 0x0 ), /* 0 */
/* 200 */ NdrFcShort( 0x0 ), /* 0 */
/* Parameter txn_in */
/* 202 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in,
by val, */
#ifdef _ALPHA_
/* 204 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
NdrFcShort( 0x8 ), /* xpp64
Stack size/offset = 8 */
#endif
/* 206 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */
/* Parameter txn_out */
/* 208 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out,
simple ref, srv alloc size=24 */
#ifdef _ALPHA_
/* 210 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
NdrFcShort( 0x20 ), /* xpp64
Stack size/offset = 32 */
#endif
/* 212 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */
/* Return value */
/* 214 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA_

```

```

/* 186 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
/* 188 */ NdrFcShort( 0x28 ), /* xpp64
Stack size/offset = 40 */
#endif
/* 218 */ 0x8, /* FC_LONG */
0x0, /* 0 */
/* Procedure CallSetComplete */
/* 220 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old
Flags: object, Oi2 */
/* 222 */ NdrFcLong( 0x0 ), /* 0 */
/* 226 */ NdrFcShort( 0x8 ), /* 8 */
/* 228 */ NdrFcShort( 0x10 ), /* ia64, xpp64 Stack size/offset = 16
*/
/* 230 */ NdrFcShort( 0x0 ), /* 0 */
/* 232 */ NdrFcShort( 0x8 ), /* 8 */
/* 234 */ 0x44, /* Oi2 Flags: has return, has ext, */
0x1, /* 1 */
/* 236 */ 0xa, /* 10 */
0x1, /* Ext
Flags: new corr desc, */
/* 238 */ NdrFcShort( 0x0 ), /* 0 */
/* 240 */ NdrFcShort( 0x0 ), /* 0 */
/* 242 */ NdrFcShort( 0x0 ), /* 0 */
/* 244 */ NdrFcShort( 0x0 ), /* 0 */
/* Return value */
/* 246 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 248 */ NdrFcShort( 0x8 ), /* ia64, xpp64 Stack size/offset = 8
*/
/* 250 */ 0x8, /* FC_LONG */
0x0, /* 0 */
0x0
}
};
static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
0,
{
NdrFcShort( 0x0 ), /* 0 */
0x12, 0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0x39e ), /* Offset= 926 (930) */
/* 6 */
0x2b, /*
0x9, /*
FC_NON_ENCAPSULATED_UNION */
FC_ULONG */

```

```

/* 8 */ 0x7, /* Corr desc: FC_USHORT */
/* 10 */ NdrFcShort( 0xffff ), /* -8 */
/* 12 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 14 */ NdrFcShort( 0x2 ), /* Offset= 2 (16) */
/* 16 */ NdrFcShort( 0x10 ), /* 16 */
/* 18 */ NdrFcShort( 0x2b ), /* 43 */
/* 20 */ NdrFcLong( 0x3 ), /* 3 */
/* 24 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 26 */ NdrFcLong( 0x11 ), /* 17 */
/* 30 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE */
/* 32 */ NdrFcLong( 0x2 ), /* 2 */
/* 36 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 38 */ NdrFcLong( 0x4 ), /* 4 */
/* 42 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT */
/* 44 */ NdrFcLong( 0x5 ), /* 5 */
/* 48 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 50 */ NdrFcLong( 0xb ), /* 11 */
/* 54 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 56 */ NdrFcLong( 0xa ), /* 10 */
/* 60 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 62 */ NdrFcLong( 0x6 ), /* 6 */
/* 66 */ NdrFcShort( 0xd6 ), /* Offset= 214 (280) */
/* 68 */ NdrFcLong( 0x7 ), /* 7 */
/* 72 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 74 */ NdrFcLong( 0x8 ), /* 8 */
/* 78 */ NdrFcShort( 0xd0 ), /* Offset= 208 (286) */
/* 80 */ NdrFcLong( 0xd ), /* 13 */
/* 84 */ NdrFcShort( 0xe4 ), /* Offset= 228 (312) */
/* 86 */ NdrFcLong( 0x9 ), /* 9 */
/* 90 */ NdrFcShort( 0xf0 ), /* Offset= 240 (330) */
/* 92 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 96 */ NdrFcShort( 0xfc ), /* Offset= 252 (348) */
/* 98 */ NdrFcLong( 0x24 ), /* 36 */
/* 102 */ NdrFcShort( 0x2f4 ), /* Offset= 756 (858) */
/* 104 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 108 */ NdrFcShort( 0x2ee ), /* Offset= 750 (858) */
/* 110 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 114 */ NdrFcShort( 0x2ec ), /* Offset= 748 (862) */
/* 116 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 120 */ NdrFcShort( 0x2ea ), /* Offset= 746 (866) */
/* 122 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 126 */ NdrFcShort( 0x2e8 ), /* Offset= 744 (870) */
/* 128 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 132 */ NdrFcShort( 0x2e6 ), /* Offset= 742 (874) */
/* 134 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 138 */ NdrFcShort( 0x2e4 ), /* Offset= 740 (878) */
/* 140 */ NdrFcLong( 0x400b ), /* 16395 */
/* 144 */ NdrFcShort( 0x2d2 ), /* Offset= 722 (866) */
/* 146 */ NdrFcLong( 0x400a ), /* 16394 */
/* 150 */ NdrFcShort( 0x2d0 ), /* Offset= 720 (870) */
/* 152 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 156 */ NdrFcShort( 0x2d6 ), /* Offset= 726 (882) */
/* 158 */ NdrFcLong( 0x4007 ), /* 16391 */

```

```

/* 162 */ NdrFcShort( 0x2cc ), /* Offset= 716 (878) */
/* 164 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 168 */ NdrFcShort( 0x2ce ), /* Offset= 718 (886) */
/* 170 */ NdrFcLong( 0x400d ), /* 16397 */
/* 174 */ NdrFcShort( 0x2cc ), /* Offset= 716 (890) */
/* 176 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 180 */ NdrFcShort( 0x2ca ), /* Offset= 714 (894) */
/* 182 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 186 */ NdrFcShort( 0x2c8 ), /* Offset= 712 (898) */
/* 188 */ NdrFcLong( 0x400c ), /* 16396 */
/* 192 */ NdrFcShort( 0x2c6 ), /* Offset= 710 (902) */
/* 194 */ NdrFcLong( 0x10 ), /* 16 */
/* 198 */ NdrFcShort( 0x8002 ), /* Simple arm type: FC_CHAR */
/* 200 */ NdrFcLong( 0x12 ), /* 18 */
/* 204 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 206 */ NdrFcLong( 0x13 ), /* 19 */
/* 210 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 212 */ NdrFcLong( 0x16 ), /* 22 */
/* 216 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 218 */ NdrFcLong( 0x17 ), /* 23 */
/* 222 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 224 */ NdrFcLong( 0xe ), /* 14 */
/* 228 */ NdrFcShort( 0x2aa ), /* Offset= 682 (910) */
/* 230 */ NdrFcLong( 0x400e ), /* 16398 */
/* 234 */ NdrFcShort( 0x2b0 ), /* Offset= 688 (922) */
/* 236 */ NdrFcLong( 0x4010 ), /* 16400 */
/* 240 */ NdrFcShort( 0x2ae ), /* Offset= 686 (926) */
/* 242 */ NdrFcLong( 0x4012 ), /* 16402 */
/* 246 */ NdrFcShort( 0x26c ), /* Offset= 620 (866) */
/* 248 */ NdrFcLong( 0x4013 ), /* 16403 */
/* 252 */ NdrFcShort( 0x26a ), /* Offset= 618 (870) */
/* 254 */ NdrFcLong( 0x4016 ), /* 16406 */
/* 258 */ NdrFcShort( 0x264 ), /* Offset= 612 (870) */
/* 260 */ NdrFcLong( 0x4017 ), /* 16407 */
/* 264 */ NdrFcShort( 0x25e ), /* Offset= 606 (870) */
/* 266 */ NdrFcLong( 0x0 ), /* 0 */
/* 270 */ NdrFcShort( 0x0 ), /* Offset= 0 (270) */
/* 272 */ NdrFcLong( 0x1 ), /* 1 */
/* 276 */ NdrFcShort( 0x0 ), /* Offset= 0 (276) */
/* 278 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (277) */
/* 280 */

FC_STRUCT */
/* 282 */ NdrFcShort( 0x8 ),
/* 284 */ 0xb,

FC_END */
/* 286 */

/* 288 */ NdrFcShort( 0xe ),
/* 290 */

```

```

FC_CARRAY */
/* 292 */ NdrFcShort( 0x2 ),
/* 294 */ 0x9,

/* 296 */ NdrFcShort( 0xffff ),
/* 298 */ NdrFcShort( 0x1 ),
/* 300 */ 0x6,

FC_END */
/* 302 */

FC_CSTRUCT */
/* 304 */ NdrFcShort( 0x8 ),
/* 306 */ NdrFcShort( 0xffffffff ),
/* 308 */ 0x8,

FC_LONG */
/* 310 */ 0x5c,

FC_END */
/* 312 */

*/

FC_CONSTANT_IID */
/* 314 */ NdrFcLong( 0x0 ),
/* 318 */ NdrFcShort( 0x0 ),
/* 320 */ NdrFcShort( 0x0 ),
/* 322 */ 0xc0,

/* 324 */ 0x0,

/* 326 */ 0x0,

/* 328 */ 0x0,

/* 330 */

*/

FC_CONSTANT_IID */
/* 332 */ NdrFcLong( 0x20400 ),
/* 336 */ NdrFcShort( 0x0 ),
/* 338 */ NdrFcShort( 0x0 ),
/* 340 */ 0xc0,

/* 342 */ 0x0,

/* 344 */ 0x0,

/* 346 */ 0x0,

```

```

0x1b,
0x1,
/* 2 */
/* Corr desc: FC_ULONG */
0x0,
/* -4 */
/* Corr flags: early, */
/* FC_SHORT */
0x5b,

0x17,

0x3,
/* 8 */
/* Offset= -16 (290) */
/* FC_LONG */
0x8,
/* FC_PAD */
0x5b,

0x2f,
/* FC_IP

0x5a,
/*

0x0 */
/* 0 */
/* 0 */
/* 0 */
0x0,
/* 0 */
0x0,
/* 0 */
0x46,
/* 70 */

0x2f,
/* FC_IP

0x5a,
/*

/* 430 */ NdrFcShort( 0x0 ),
/* 432 */ NdrFcShort( 0x1 ),
/* 434 */ NdrFcLong( 0xffffffff ),
/* 438 */ NdrFcShort( 0x0 ),
/* 440 */

/* 442 */ NdrFcShort( 0xffffffff74 ),
/* 444 */ 0x5c,

FC_END */
/* 446 */

```

FC_BOGUS_STRUCT */	0x1a,	/*	/* 502 */ NdrFcShort(0x0),	0x0,	/* 3 */	/* 566 */ NdrFcShort(0x0),	/* Offset= 6 (572) */
	0x3,	/* 3 */	/* 504 */ 0x19,	/* Corr desc: field pointer,		/* 568 */ 0x8,	/* FC_LONG */
			FC_ULONG */				0x39,
/* 448 */ NdrFcShort(0x10),	/* 16 */		/* 506 */ NdrFcShort(0x0),	0x0,	/* */	FC_ALIGNM8 */	
/* 450 */ NdrFcShort(0x0),	/* 0 */		/* 508 */ NdrFcShort(0x1),	/* 0 */		/* 570 */ 0x36,	/* FC_POINTER */
/* 452 */ NdrFcShort(0x6),	/* Offset= 6 (458) */		/* 510 */ NdrFcLong(0xffffffff),	/* Corr flags: early, */			0x5b,
/* 454 */ 0x8,	/* FC_LONG */		/* 514 */ NdrFcShort(0x0),	/* -1 */		FC_END */	
	0x39,	/*	/* 516 */ 0x4c,	/* Corr flags: */		/* 572 */	
FC_ALIGNM8 */				/* FC_EMBEDDED_COMPLEX */			0x11, 0x0, /* FC_RP */
/* 456 */ 0x36,	/* FC_POINTER */		/* 518 */ NdrFcShort(0xffffffff44),	/* Offset= -188 (330) */		/* 574 */ NdrFcShort(0xffffffffdc),	/* Offset= -36 (538) */
	0x5b,	/*	/* 520 */ 0x5c,	/* FC_PAD */		/* 576 */	
FC_END */				0x5b,	/*		0x2f,
/* 458 */			FC_END */				/* FC_IP
			/* 522 */				0x5a,
/* 460 */ NdrFcShort(0xffffffffdc),	0x11, 0x0, /* FC_RP */			0x1a,	/*	FC_CONSTANT_IID */	
/* 462 */	/* Offset= -36 (424) */					/* 578 */ NdrFcLong(0x2f),	/* 47 */
			FC_BOGUS_STRUCT */	0x3,	/* 3 */	/* 582 */ NdrFcShort(0x0),	/* 0 */
	0x21,	/*				/* 584 */ NdrFcShort(0x0),	/* 0 */
FC_BOGUS_ARRAY */			/* 524 */ NdrFcShort(0x10),	/* 16 */		/* 586 */ 0xc0,	/* 192 */
	0x3,	/* 3 */	/* 526 */ NdrFcShort(0x0),	/* 0 */			0x0,
/* 464 */ NdrFcShort(0x0),	/* 0 */		/* 528 */ NdrFcShort(0x6),	/* Offset= 6 (534) */		/* 588 */ 0x0,	/* 0 */
/* 466 */ 0x19,	/* Corr desc: field pointer,		/* 530 */ 0x8,	/* FC_LONG */			0x0,
FC_ULONG */				0x39,	/*	/* 590 */ 0x0,	/* 0 */
	0x0,	/* */	FC_ALIGNM8 */			/* 592 */ 0x0,	/* 0 */
/* 468 */ NdrFcShort(0x0),	/* 0 */		/* 532 */ 0x36,	/* FC_POINTER */			0x0,
/* 470 */ NdrFcShort(0x1),	/* Corr flags: early, */			0x5b,	/*	/* 594 */	0x46,
/* 472 */ NdrFcLong(0xffffffff),	/* -1 */		FC_END */				/* 70 */
/* 476 */ NdrFcShort(0x0),	/* Corr flags: */		/* 534 */				0x1b,
/* 478 */ 0x4c,	/* FC_EMBEDDED_COMPLEX */					FC_CARRAY */	
	0x0,	/* 0 */	/* 536 */ NdrFcShort(0xffffffffdc),	0x11, 0x0, /* FC_RP */			0x0,
/* 480 */ NdrFcShort(0xffffffff58),	/* Offset= -168 (312) */		/* 538 */	/* Offset= -36 (500) */		/* 596 */ NdrFcShort(0x1),	/* 1 */
/* 482 */ 0x5c,	/* FC_PAD */					/* 598 */ 0x19,	/* Corr desc: field pointer,
	0x5b,	/*	FC_BOGUS_ARRAY */	0x21,	/*	FC_ULONG */	
FC_END */							0x0,
/* 484 */			/* 540 */ NdrFcShort(0x0),	0x3,	/* 3 */	/* 600 */ NdrFcShort(0x4),	/* 4 */
	0x1a,	/*	/* 542 */ 0x19,	/* Corr desc: field pointer,		/* 602 */ NdrFcShort(0x1),	/* Corr flags: early, */
FC_BOGUS_STRUCT */			FC_ULONG */			/* 604 */ 0x1,	/* FC_BYTE */
	0x3,	/* 3 */		0x0,	/* */		0x5b,
/* 486 */ NdrFcShort(0x10),	/* 16 */		/* 544 */ NdrFcShort(0x0),	/* 0 */		FC_END */	
/* 488 */ NdrFcShort(0x0),	/* 0 */		/* 546 */ NdrFcShort(0x1),	/* 0 */		/* 606 */	
/* 490 */ NdrFcShort(0x6),	/* Offset= 6 (496) */		/* 548 */ NdrFcLong(0xffffffff),	/* Corr flags: early, */			0x1a,
/* 492 */ 0x8,	/* FC_LONG */		/* 552 */ NdrFcShort(0x0),	/* -1 */		FC_BOGUS_STRUCT */	
	0x39,	/*	/* 554 */	/* Corr flags: */			0x3,
FC_ALIGNM8 */						/* 608 */ NdrFcShort(0x18),	/* 24 */
/* 494 */ 0x36,	/* FC_POINTER */		/* 556 */ NdrFcShort(0x176),	0x12, 0x0, /* FC_UP */		/* 610 */ NdrFcShort(0x0),	/* 0 */
	0x5b,	/*	/* 558 */ 0x5c,	/* Offset= 374 (930) */		/* 612 */ NdrFcShort(0xc),	/* Offset= 12 (624) */
FC_END */				/* FC_PAD */		/* 614 */ 0x8,	/* FC_LONG */
/* 496 */			FC_END */	0x5b,	/*		0x8,
			/* 560 */			FC_LONG */	
/* 498 */ NdrFcShort(0xffffffffdc),	0x11, 0x0, /* FC_RP */			0x1a,	/*	/* 616 */ 0x4c,	/* FC_EMBEDDED_COMPLEX */
/* 500 */	/* Offset= -36 (462) */		FC_BOGUS_STRUCT */				0x0,
				0x3,	/* 3 */	/* 618 */ NdrFcShort(0xffffffffd6),	/* Offset= -42 (576) */
FC_BOGUS_ARRAY */	0x21,	/*	/* 562 */ NdrFcShort(0x10),	/* 16 */		/* 620 */ 0x39,	/* FC_ALIGNM8 */

FC_POINTER */	0x36,	/*	FC_STRUCT */	0x15,	/*	/* 720 */ NdrFcShort(0x0),	/* Offset= 6 (728) */
/* 622 */ 0x5c,	/* FC_PAD */		/* 674 */ NdrFcShort(0x10),	0x3,	/* 3 */	/* 724 */ 0x8,	/* FC_LONG */
FC_END */	0x5b,	/*	/* 676 */ 0x8,	/* 16 */			0x39,
/* 624 */				/* FC_LONG */		FC_ALIGNM8 */	/* FC_POINTER */
				0x6,	/*	/* 726 */ 0x36,	0x5b,
/* 626 */ NdrFcShort(0xffffffe0),	0x12, 0x0, /* FC_UP */		FC_SHORT */	/* FC_SHORT */		FC_END */	/*
/* 628 */	/* Offset= -32 (594) */		/* 678 */ 0x6,	0x4c,	/*	/* 728 */	
			FC_EMBEDDED_COMPLEX */				0x12, 0x0, /* FC_UP */
	0x21,	/*	/* 680 */ 0x0,	/* 0 */		/* 730 */ NdrFcShort(0xffffffe6),	/* Offset= -26 (704) */
FC_BOGUS_ARRAY */			Offset= -15 (666) */	NdrFcShort(0xffffffff),	/*	/* 732 */	
	0x3,	/* 3 */		0x5b,	/*	FC_CARRAY */	0x1b,
/* 630 */ NdrFcShort(0x0),	/* 0 */		FC_END */				/* 1 */
/* 632 */ 0x19,	/* Corr desc: field pointer,		/* 684 */	0x1a,	/*	/* 734 */ NdrFcShort(0x2),	/* 2 */
FC_ULONG */			FC_BOGUS_STRUCT */			/* 736 */ 0x19,	/* Corr desc: field pointer,
	0x0,	/* */		0x3,	/* 3 */	FC_ULONG */	
/* 634 */ NdrFcShort(0x0),	/* 0 */		/* 686 */ NdrFcShort(0x20),	/* 32 */		/* 738 */ NdrFcShort(0x0),	0x0,
/* 636 */ NdrFcShort(0x1),	/* Corr flags: early, */		/* 688 */ NdrFcShort(0x0),	/* 0 */		/* 740 */ NdrFcShort(0x1),	/* 0 */
/* 638 */ NdrFcLong(0xffffffff),	/* -1 */		/* 690 */ NdrFcShort(0xa),	/* Offset= 10 (700) */		/* 742 */ 0x6,	/* Corr flags: early, */
/* 642 */ NdrFcShort(0x0),	/* Corr flags: */		/* 692 */ 0x8,	/* FC_LONG */			/* FC_SHORT */
/* 644 */				0x39,	/*	FC_END */	0x5b,
	0x12, 0x0, /* FC_UP */		FC_ALIGNM8 */			/* 744 */	
/* 646 */ NdrFcShort(0xffffffd8),	/* Offset= -40 (606) */		/* 694 */ 0x36,	/* FC_POINTER */			0x1a,
/* 648 */ 0x5c,	/* FC_PAD */	/*	FC_EMBEDDED_COMPLEX */	0x4c,	/*	FC_BOGUS_STRUCT */	
	0x5b,		/* 696 */ 0x0,	/* 0 */		/* 746 */ NdrFcShort(0x10),	0x3,
FC_END */				NdrFcShort(0xffffffff7),	/*	/* 748 */ NdrFcShort(0x0),	/* 16 */
/* 650 */	0x1a,	/*	Offset= -25 (672) */	0x5b,	/*	/* 750 */ NdrFcShort(0x6),	/* 0 */
FC_BOGUS_STRUCT */			FC_END */			/* 752 */ 0x8,	/* Offset= 6 (756) */
	0x3,	/* 3 */	/* 700 */	0x11, 0x0, /* FC_RP */			/* FC_LONG */
/* 652 */ NdrFcShort(0x10),	/* 16 */		/* 702 */ NdrFcShort(0xffffffff10),	/* Offset= -240 (462) */		FC_ALIGNM8 */	0x39,
/* 654 */ NdrFcShort(0x0),	/* 0 */		/* 704 */	0x1b,	/*	/* 754 */ 0x36,	/* FC_POINTER */
/* 656 */ NdrFcShort(0x6),	/* Offset= 6 (662) */		FC_CARRAY */			FC_END */	0x5b,
/* 658 */ 0x8,	/* FC_LONG */	/*	/* 706 */ NdrFcShort(0x1),	0x0,	/* 0 */	/* 756 */	
	0x39,		/* 708 */ 0x19,	/* 1 */	/*	/* 758 */ NdrFcShort(0xffffffe6),	0x12, 0x0, /* FC_UP */
FC_ALIGNM8 */	/* FC_POINTER */		FC_ULONG */	/* Corr desc: field pointer,		/* 760 */	/* Offset= -26 (732) */
/* 660 */ 0x36,	0x5b,	/*				FC_CARRAY */	0x1b,
FC_END */			/* 710 */ NdrFcShort(0x0),	0x0,	/* *		/*
/* 662 */			/* 712 */ NdrFcShort(0x1),	/* 0 */		/* 762 */ NdrFcShort(0x4),	0x3,
	0x11, 0x0, /* FC_RP */	/*	/* 714 */ 0x1,	/* Corr flags: early, */		/* 764 */ 0x19,	/* 4 */
/* 664 */ NdrFcShort(0xfffffdc),	/* Offset= -36 (628) */			/* FC_BYTE */		FC_ULONG */	/* Corr desc: field pointer,
/* 666 */				0x5b,	/*		
FC_SMFARRAY */	0x1d,	/*	FC_END */			/* 766 */ NdrFcShort(0x0),	0x0,
	0x0,	/* 0 */	/* 716 */	0x1a,	/*	/* 768 */ NdrFcShort(0x1),	/* 0 */
/* 668 */ NdrFcShort(0x8),	/* 8 */		FC_BOGUS_STRUCT */			/* 770 */ 0x8,	/* Corr flags: early, */
/* 670 */ 0x2,	/* FC_CHAR */	/*	/* 718 */ NdrFcShort(0x10),	0x3,	/* 3 */		/* FC_LONG */
	0x5b,			/* 16 */		FC_END */	0x5b,
FC_END */						/* 772 */	/*
/* 672 */							

FC_BOGUS_STRUCT */	0x1a,	/*	/* 822 */ 0x5c,	/* 822 */ 0x5c,	/* 822 */ 0x5c,	/* 870 */ 0x12, 0x8,	/* FC_UP
/* 774 */ NdrFcShort(0x10),	/* 16 */	/* 3 */	FC_END */	FC_END */	FC_END */	[simple_pointer] */	/* FC_LONG */
/* 776 */ NdrFcShort(0x0),	/* 0 */		/* 824 */	/* 824 */	/* 824 */	/* 872 */ 0x8,	/* FC_LONG */
/* 778 */ NdrFcShort(0x6),	/* Offset= 6 (784) */		FC_CARRAY */	FC_CARRAY */	FC_CARRAY */	FC_PAD */	/* FC_LONG */
/* 780 */ 0x8,	/* FC_LONG */	/*	/* 826 */ NdrFcShort(0x8),	/* 826 */ NdrFcShort(0x8),	/* 826 */ NdrFcShort(0x8),	/* 874 */	/* FC_UP
FC_ALIGNM8 */	0x39,	/*	/* 828 */ 0x7,	/* 828 */ 0x7,	/* 828 */ 0x7,	[simple_pointer] */	/* FC_UP
/* 782 */ 0x36,	/* FC_POINTER */		/* 830 */ NdrFcShort(0xffc8),	/* 830 */ NdrFcShort(0xffc8),	/* 830 */ NdrFcShort(0xffc8),	/* 876 */ 0xa,	/* FC_UP
FC_END */	0x5b,	/*	/* 832 */ NdrFcShort(0x1),	/* 832 */ NdrFcShort(0x1),	/* 832 */ NdrFcShort(0x1),	/* 878 */	/* FC_UP
/* 784 */			/* 834 */ 0x4c,	/* 834 */ 0x4c,	/* 834 */ 0x4c,	FC_PAD */	/* FC_UP
/* 786 */ NdrFcShort(0xfffffe6),	0x12, 0x0, /* FC_UP */		/* 836 */ NdrFcShort(0xfffffec),	/* 836 */ NdrFcShort(0xfffffec),	/* 836 */ NdrFcShort(0xfffffec),	[simple_pointer] */	/* FC_UP
/* 788 */	/* Offset= -26 (760) */		/* 838 */ 0x5c,	/* 838 */ 0x5c,	/* 838 */ 0x5c,	/* 880 */ 0xc,	/* FC_UP
FC_CARRAY */	0x1b,	/*	FC_END */	FC_END */	FC_END */	/* 882 */	/* FC_UP
/* 790 */ NdrFcShort(0x8),	0x7,	/* 7 */	/* 840 */	/* 840 */	/* 840 */	FC_PAD */	/* FC_UP
/* 792 */ 0x19,	/* 8 */		FC_BOGUS_STRUCT */	FC_BOGUS_STRUCT */	FC_BOGUS_STRUCT */	/* 884 */ NdrFcShort(0xffffda4),	/* FC_UP
FC_ULONG */	/* Corr desc: field pointer,		/* 842 */ NdrFcShort(0x38),	/* 842 */ NdrFcShort(0x38),	/* 842 */ NdrFcShort(0x38),	/* 886 */	/* FC_UP
/* 794 */ NdrFcShort(0x0),	0x0,	/* *	/* 844 */ NdrFcShort(0xfffffec),	/* 844 */ NdrFcShort(0xfffffec),	/* 844 */ NdrFcShort(0xfffffec),	[pointer_deref] */	/* FC_UP
/* 796 */ NdrFcShort(0x1),	/* 0 */		/* 846 */ NdrFcShort(0x0),	/* 846 */ NdrFcShort(0x0),	/* 846 */ NdrFcShort(0x0),	/* 888 */ NdrFcShort(0xffffda6),	/* FC_UP
/* 798 */ 0xb,	/* Corr flags: early, */		/* 848 */ 0x6,	/* 848 */ 0x6,	/* 848 */ 0x6,	/* 890 */	/* FC_UP
FC_END */	/* FC_HYPER */		FC_SHORT */	FC_SHORT */	FC_SHORT */	[pointer_deref] */	/* FC_UP
/* 800 */	0x5b,	/*	/* 850 */ 0x38,	/* 850 */ 0x38,	/* 850 */ 0x38,	/* 892 */ NdrFcShort(0xffffdbc),	/* FC_UP
FC_BOGUS_STRUCT */	0x1a,	/*	/* 852 */ 0x8,	/* 852 */ 0x8,	/* 852 */ 0x8,	/* 894 */	/* FC_UP
/* 802 */ NdrFcShort(0x10),	0x3,	/* 3 */	FC_LONG */	FC_LONG */	FC_LONG */	[pointer_deref] */	/* FC_UP
/* 804 */ NdrFcShort(0x0),	/* 16 */		/* 854 */ 0x4,	/* 854 */ 0x4,	/* 854 */ 0x4,	/* 896 */ NdrFcShort(0xffffdca),	/* FC_UP
/* 806 */ NdrFcShort(0x6),	/* 0 */		FC_EMBEDDED_COMPLEX */	FC_EMBEDDED_COMPLEX */	FC_EMBEDDED_COMPLEX */	/* 898 */	/* FC_UP
/* 808 */ 0x8,	/* Offset= 6 (812) */		Offset= -499 (356) */	Offset= -499 (356) */	Offset= -499 (356) */	[pointer_deref] */	/* FC_UP
FC_ALIGNM8 */	/* FC_LONG */	/*	FC_END */	FC_END */	FC_END */	/* 900 */ NdrFcShort(0xffffdd8),	/* FC_UP
/* 810 */ 0x36,	0x39,	/*	/* 858 */	/* 858 */	/* 858 */	/* 902 */	/* FC_UP
FC_END */	/* FC_POINTER */		/* 860 */ NdrFcShort(0xfffff02),	/* 860 */ NdrFcShort(0xfffff02),	/* 860 */ NdrFcShort(0xfffff02),	[pointer_deref] */	/* FC_UP
/* 812 */	0x5b,	/*	/* 862 */	/* 862 */	/* 862 */	/* 904 */ NdrFcShort(0x2),	/* FC_UP
/* 814 */ NdrFcShort(0xfffffe6),	0x12, 0x0, /* FC_UP */		[simple_pointer] */	[simple_pointer] */	[simple_pointer] */	/* 906 */	/* FC_UP
/* 816 */	/* Offset= -26 (788) */		/* 864 */ 0x1,	/* 864 */ 0x1,	/* 864 */ 0x1,	/* 908 */ NdrFcShort(0x16),	/* FC_UP
FC_STRUCT */	0x15,	/*	FC_PAD */	FC_PAD */	FC_PAD */	/* 910 */	/* FC_UP
/* 818 */ NdrFcShort(0x8),	0x3,	/* 3 */	/* 866 */	/* 866 */	/* 866 */	FC_STRUCT */	/* FC_UP
/* 820 */ 0x8,	/* 8 */		[simple_pointer] */	[simple_pointer] */	[simple_pointer] */	/* 912 */ NdrFcShort(0x10),	/* FC_UP
FC_LONG */	/* FC_LONG */	/*	/* 868 */ 0x6,	/* 868 */ 0x6,	/* 868 */ 0x6,	/* 914 */ 0x6,	/* FC_UP
	0x8,		FC_PAD */	FC_PAD */	FC_PAD */	FC_BYTE */	/* FC_UP
						/* 916 */ 0x1,	/* FC_UP

```

0x38, /*
FC_ALIGNM4 */
/* 918 */ 0x8,
/* FC_LONG */
0x39, /*
FC_ALIGNM8 */
/* 920 */ 0xb,
/* FC_HYPER */
0x5b, /*
FC_END */
/* 922 */
0x12, 0x0, /* FC_UP */
/* 924 */ NdrFcShort( 0xffffffff ), /* Offset= -14 (910) */
/* 926 */
0x12, 0x8, /* FC_UP
[simple_pointer] */
/* 928 */ 0x2,
/* FC_CHAR */
0x5c, /*
FC_PAD */
/* 930 */
0x1a, /*
FC_BOGUS_STRUCT */
0x7, /* 7 */
/* 932 */ NdrFcShort( 0x20 ), /* 32 */
/* 934 */ NdrFcShort( 0x0 ), /* 0 */
/* 936 */ NdrFcShort( 0x0 ), /* Offset= 0 (936) */
/* 938 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 940 */ 0x6,
/* FC_SHORT */
0x6, /*
FC_SHORT */
0x6, /*
FC_SHORT */
/* 942 */ 0x6,
/* FC_SHORT */
/* 944 */ 0x4c,
/* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 946 */ NdrFcShort( 0xffffffff54 ), /* Offset= -940 (6) */
/* 948 */ 0x5c,
/* FC_PAD */
0x5b, /*
FC_END */
/* 950 */ 0xb4,
/* FC_USER_MARSHAL */
0x83, /* 131 */
/* 952 */ NdrFcShort( 0x0 ), /* 0 */
/* 954 */ NdrFcShort( 0x18 ), /* 24 */
/* 956 */ NdrFcShort( 0x0 ), /* 0 */
/* 958 */ NdrFcShort( 0xffffffff44 ), /* Offset= -956 (2) */
/* 960 */
0x11, 0x4, /* FC_RP
[allocated_on_stack] */
/* 962 */ NdrFcShort( 0x6 ), /* Offset= 6 (968) */
/* 964 */
0x13, 0x0, /* FC_OP */
/* 966 */ NdrFcShort( 0xffffffffdc ), /* Offset= -36 (930) */

```

```

/* 968 */ 0xb4,
/* 970 */ NdrFcShort( 0x0 ),
/* 972 */ NdrFcShort( 0x18 ),
/* 974 */ NdrFcShort( 0x0 ),
/* 976 */ NdrFcShort( 0xffffffff4 ), /* Offset= -12 (964) */
0x0
}
};

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

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

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

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

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{
if(!_tpcc_com_ps_CHECK_IID(0))
{
*pIndex = 0;
return 1;
}

return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo =
{
(PCInterfaceProxyVtblList *) &_tpcc_com_ps_ProxyVtblList,
(PCInterfaceStubVtblList *) &_tpcc_com_ps_StubVtblList,
(const PCInterfaceName *) &_tpcc_com_ps_InterfaceNamesList,
0, // no delegation
&_tpcc_com_ps_IID_Lookup,
1,
2,

```

```

0, /* File1 of [async_uuid] interfaces */
0, /* Filler2 */
0 /* Filler3 */
};

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

```

tpcc_com_ps\src\dlldata.c

```

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

DO NOT ALTER THIS FILE

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

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

*****/

#include <rpcproxy.h>

#ifdef __cplusplus
extern "C" {
#endif

EXTERN_PROXY_FILE( tpcc_com_ps )

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

DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )

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

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

```


Appendix B Database Load

Setup.cmd

```
@echo off

set DoDBCREATE=FALSE
set DoBULKINSERT=TRUE
set DoCLEANUP=TRUE
set DoBACKUP=FALSE

rem
rem Modify the following parameters for your configuration
rem

set HOMEDRIVE=G:
set HOMEDIR=\hptpcc
set OUTPUTDRIVE=G:
set OUTPUTDIR=\hptpcc\Output
set THREADS=32
set WAREHOUSES=60000
set DB=tpcc
set PASSWD=ssdl

set HOMEPATH=%HOMEDRIVE%%HOMEDIR%
set FILEPATH=%FILEDRIVE%%FILEDIR%

if %PROCESSOR_ARCHITECTURE% == x86 (set
EXECUTABLEPATH=%HOMEPATH%\Executables\x86
) else set EXECUTABLEPATH=%HOMEPATH%\Executables\ipf

%HOMEDRIVE%
cd %HOMEDIR%

if '%1' == 'BULKINSERTn' goto :BULKINSERTn
if '%1' == 'DBCREATE' goto :DBCREATE

echo Checking for existence of HOMEDIR and OUTPUTDIR
if NOT EXIST %HOMEDRIVE%%HOMEDIR% goto :NO_HOMEDIR
if NOT EXIST %OUTPUTDRIVE%%OUTPUTDIR% goto :NO_OUTPUTDIR

echo Finding next output directory in %OUTPUTDRIVE%%OUTPUTDIR%
set OUTPUTNUMBER=1
:OUTPUTLOOP
```

```
if NOT EXIST %OUTPUTDRIVE%%OUTPUTDIR%\%OUTPUTNUMBER%
goto :OUTPUTLOOPEND
set /a OUTPUTNUMBER=%OUTPUTNUMBER%+1
goto :OUTPUTLOOP
:OUTPUTLOOPEND
set OUTPUTPATH=%OUTPUTDRIVE%%OUTPUTDIR%\%OUTPUTNUMBER%
echo Output will be found at %OUTPUTPATH%
mkdir %OUTPUTPATH%

:DBCREATE
if NOT '%DoDBCREATE%' == 'TRUE' goto :BULKINSERT
rem
rem DBCREATE invokes the file CreateDatabase.sql
rem
echo Starting database creation
time /t >> %OUTPUTPATH%\time.txt
echo Starting database creation >> %OUTPUTPATH%\time.txt
%HOMEDRIVE%
cd %HOMEDIR%\Database
osql -E -i CreateDatabase.sql -o %OUTPUTPATH%\CreateDatabase.out -b
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"sp_dboption %DB%, 'torn', FALSE" -b
if ERRORLEVEL 1 goto :ERROR_EXIT
if '%1' == 'DBCREATE' %EXECUTABLEPATH%\Semaphore -release
echo DBCREATE done %OUTPUTPATH%\time.txt
time /t %OUTPUTPATH%\time.txt
if '%1' == 'DBCREATE' goto :EOF

:BULKINSERT
if NOT '%DoBULKINSERT%' == 'TRUE' goto :CLEANUP
rem
rem BULKINSERT loads each table through a set of concurrent bulk inserts
rem After each table is loaded, the table is indexed
rem

echo Starting Bulk Inserts
%HOMEDRIVE%
cd %HOMEDIR%\TablesAndIndexes

osql -E -d%DB% -i%HOMEPATH%\Database\dbopt1.sql -o
%OUTPUTPATH%\dbopt1.out
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"sp_configure 'show advanced options',1" -b
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"reconfigure with override" -b
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"sp_configure 'max degree',0" -b
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"reconfigure with override" -b
if ERRORLEVEL 1 goto :ERROR_EXIT
```

```
echo Dropping and Re-Creating Tables
osql -E -d%DB% -i%HOMEPATH%\CreateTables.sql -o %OUTPUTPATH%\CreateTables.out -b
if ERRORLEVEL 1 goto :ERROR_EXIT

rem Special case for Item, since it is only one file
echo Starting create indexes for Item
osql -E -d%DB% -
i%HOMEPATH%\TablesAndIndexes\NewCreateITEMIndexes.sql -
o%OUTPUTPATH%\NewCreateitemIndexes.out
if ERRORLEVEL 1 goto :ERROR_EXIT
echo Starting bulk inserts for Item
%EXECUTABLEPATH%\stockldr\tpccldr -s1 -W10 -titem -Pssdl -i0 -
f%OUTPUTPATH%\loaditem.out -L%OUTPUTPATH%\
if ERRORLEVEL 1 goto :ERROR_EXIT

echo Starting bulk inserts of item customer warehouse orders

for %i in (warehouse customer orders) do call :BULKINSERTi %i
if ERRORLEVEL 1 goto :ERROR_EXIT

goto :CLEANUP

:BULKINSERTi
echo Starting bulk inserts for %i
for /l %j in (1,1,%THREADS%) do start cmd /C
%HOMEPATH%\SetupOldLoader BULKINSERTn %i %j
%EXECUTABLEPATH%\Semaphore -wait %THREADS%

echo Starting create indexes for %i
osql -E -d%DB% -
i%HOMEPATH%\TablesAndIndexes\NewCreate%1Indexes.sql -o
%OUTPUTPATH%\NewCreate%1Indexes.out
if ERRORLEVEL 1 goto :ERROR_EXIT
goto :EOF

:BULKINSERTn
set /a REM=WAREHOUSES %* THREADS
set /a WAREHOUSEPERTHREAD=WAREHOUSES / THREADS
set /a TEMP=%3-1
set /a STARTWAREHOUSE=TEMP%*WAREHOUSEPERTHREAD%
if %3 leq %REM% set /a
WAREHOUSEPERTHREAD=WAREHOUSEPERTHREAD%+1
if %3 leq %REM% set /a
STARTWAREHOUSE=STARTWAREHOUSE%+%3%
if %3 gtr %REM% set /a
STARTWAREHOUSE=STARTWAREHOUSE%+%REM%+1
set /a
ENDWAREHOUSE=STARTWAREHOUSE%+%WAREHOUSEPERTHREAD%-1
%EXECUTABLEPATH%\stockldr\tpccldr -t%2 -Pssdl -
s%STARTWAREHOUSE% -W%ENDWAREHOUSE% -i0 -
f%OUTPUTPATH%\load%2%STARTWAREHOUSE%.out -L%OUTPUTPATH%\
```



```
%EXECUTABLEPATH%\Semaphore -release
goto :EOF

:CLEANUP
if NOT '%DoCLEANUP%' == 'TRUE' goto :BACKUP
rem
rem CLEANUP sets statistics and lock options
rem

echo Setting Cleanup Options

%HOMEDRIVE%
cd %HOMEDIR%\Database
osql -E -d%DB% -i%HOMEPATH%\Database\dbopt2.sql -o
%OUTPUTPATH%\dbopt2.out
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -d%DB% -Q'sp_dboption '%DB%', 'auto create statistics', 'OFF' -o
%OUTPUTPATH%\AutoCreateStats.out
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -d%DB% -Q'sp_dboption '%DB%', 'auto update statistics', 'OFF' -o
%OUTPUTPATH%\AutoUpdateStats.out
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q'sp_configure 'max degree', 1 -b
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q'reconfigure with override' -b
if ERRORLEVEL 1 goto :ERROR_EXIT

cd %HOMEDIR%\StoredProcedures
for %%i in (Create*Proc.sql) do osql -E -d%DB% -
i%HOMEPATH%\StoredProcedures\%%i -o %OUTPUTPATH%\%%i.out -b

osql -E -Q'sp_dboption %DB%, 'trunc', FALSE" -b
>%OUTPUTPATH%\dboptionTrunOFF.out
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q'sp_dboption %DB%, 'select', FALSE" -b
>%OUTPUTPATH%\dboptionSelOFF.out
if ERRORLEVEL 1 goto :ERROR_EXIT

:BACKUP
rem
rem BACKUP is the final step, using a script
rem

if NOT '%DoBACKUP%' == 'TRUE' goto :DONE
%HOMEDRIVE%
cd %HOMEDIR%\Database

echo Starting Backup
osql -E -d%DB% -i%HOMEPATH%\database\Backup.sql -o
%OUTPUTPATH%\Backup.out -b
```

```
if ERRORLEVEL 1 goto :ERROR_EXIT
:DONE

echo Done! Check for output in %OUTPUTPATH%
%HOMEDRIVE%
cd %HOMEDIR%

goto :EOF

:NO_HOMEDIR
echo directory %HOMEDRIVE%%HOMEDIR% does not exist
goto :ERROR_EXIT

:NO_OUTPUTDIR
echo directory %OUTPUTDRIVE%%OUTPUTDIR% does not exist
goto :ERROR_EXIT

:ERROR_EXIT
echo SetupOldLoader aborted due to errors
echo Check output in %OUTPUTPATH%
%HOMEDRIVE%
cd %HOMEDIR%
exit /B
```

tpcc.h

```
// File: TPCC.H
// Microsoft TPC-C Kit
// Ver. 4.42
// Copyright Microsoft,
// 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
// Purpose: Header file for TPC-C database loader

// Build number of TPC Benchmark Kit
#define TPCKIT_VER "4.42"

// 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 <math.h>

// ODBC headers
#include <sql.h>
#include <sqltext.h>
#include <odbc.h>

// General constants
#define MILLI 1000
#define FALSE 0
#define TRUE 1
#define UNDEF -1
#define MINPRINTASCII 32
#define MAXPRINTASCII 126
#define MAXSQLERRORS 10

// 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 ".\\load.out"
#define LOADER_LOG_PATH ".\\";
#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"
#define RETRYLIMIT 5
#define SLEEPMILLI 1000

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
        long                        num_warehouses;
        long                        batch;
        long                        verbose;

        long
        pack_size;
        char
        *loader_res_file;
        char
        *log_path;
        char
        *synch_servername;
        long
        case_sensitivity;
        long
        starting_warehouse;
        long
        build_index;
        long
        index_order;
        long
        scale_down;
        char
        *index_script_path;
        long
        retry_limit;
        long
        sleep_milli;
} TPCCCLR_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          220
#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:                TPCCCLR.C
//                                Microsoft TPC-C Kit
Ver. 4.42
//                                Copyright Microsoft,
1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
//      Purpose:            Source file for TPC-C database loader

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

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

// Functions declarations

void HandleErrorDBC(SQLHDBC hdbc1);
void CheckDataBase();
long NURand();
void LoadItem();
void LoadWarehouse();
void Stock();
void District();
void LoadCustomer();
void CustomerBufInit();
void CustomerBufLoad();
void LoadCustomerTable();
void LoadHistoryTable();
void LoadOrders();
void OrdersBufInit();
void OrdersBufLoad();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void GetPermutation();
void CheckForCommit();
void CheckForCommit_Big();
void OpenConnections();
void BuildIndex();
void FormatDate ();

```

```

// Shared memory structures

typedef struct
{
    double          ol_i_id;
    long            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;
    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;
    long            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_discount;
    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;

long            totaldberrors = 0;

ORDERS_STRUCT  orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT customer_buf[CUSTOMERS_PER_DISTRICT];
long            orders_rows_loaded;
double          new_order_rows_loaded;
double          order_line_rows_loaded;
long            history_rows_loaded;
long            customer_rows_loaded;
double          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*****\n");
printf("\n*          *\n");
printf("\n* Microsoft SQL Server          *\n");
printf("\n*          *\n");
printf("\n* TPC-C BENCHMARK KIT: Database loader\n");
printf("\n*          *\n");
printf("\n* Version %s          *\n",
TPCKIT_VER);
printf("\n*          *\n");
printf("\n*****\n");

// process command line arguments

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

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;

```

```

first_new_order = 1;
last_new_order = 30;
ORDERS_PER_DISTRICT;

}

// 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. starting warehouse
=%ld\n", (aptr->num_warehouses-aptr->starting_warehouse+1),
aptr->starting_warehouse);

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

main_time_start = (TimeNow()) / MILLI);

// start parallel load threads

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

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

if (hThread[0] == NULL)

```

```

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

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

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

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

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

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

if (hThread[2] == NULL)
{

```

```

        printf("Error, failed in creating
creating main thread = 2.\n");
        exit(-1);
    }
}

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

    hThread[3] = CreateThread(NULL,

        0,

        (LPTHREAD_START_ROUTINE)

        LoadOrders,

        NULL,

        0,

        &dwThreadID[3]);

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

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

        CloseHandle(hThread[i]);
        hThread[i] = NULL;
    }
}

main_time_end = (TimeNow() / MILLI);

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

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

```

```

fclose(floader);

SQLFreeEnv(henv);

exit(0);

return 0;
}

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

void LoadItem()
{
    long        i_id;
    long        i_im_id;
    char        i_name[I_NAME_LEN+1];
    double      i_price;
    char        i_data[I_DATA_LEN+1];
    char        name[20];
    long        time_start;
    RETCODE     rc;
    DBINT       rcint;
    char        bcphint[128];
    char        err_log_path[256];

    // Seed with unique number
    seed(1);

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

    // if build index before load
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        BuildIndex("idxitmc1");

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

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

    //rc = bcp_init(i_hdbc1, name, NULL, "logs\\item.err", DB_IN);
    strcpy(err_log_path, aptr->log_path);
    strcat(err_log_path, "item.err");
    rc = bcp_init(i_hdbc1, name, NULL, err_log_path, DB_IN);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

```

```

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        sprintf(bcphint, "tablock, 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("idxitmc1");
}

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

void LoadWarehouse()
{
    long w_id;
    char w_name[W_NAME_LEN+1];
    char w_street_1[ADDRESS_LEN+1];
    char w_street_2[ADDRESS_LEN+1];
    char w_city[ADDRESS_LEN+1];
    char w_state[STATE_LEN+1];
    char w_zip[ZIP_LEN+1];
    double w_tax;
    double w_ytd;
    char name[20];
    long time_start;
    RETCODE rc;
    DBINT rcint;
    char bcphint[128];
    char err_log_path[256];

```

```

    // Seed with unique number
    seed(2);

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

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

    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");

    // strcpy(err_log_path, aptr->log_path);
    // strcat(err_log_path, "whouse.err");
    sprintf(err_log_path, "%swhouse%d.err", aptr->log_path, aptr-
>starting_warehouse);
    printf("warehouse error file =%s\n", err_log_path);
    rc = bcp_init(w_hdbc1, name, NULL, err_log_path, DB_IN);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcphint, "tablock, 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, SQLINT4, 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 = (long)aptr->starting_warehouse; w_id <=
(long)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("idxward");

//stock_rows_loaded = 0;
stock_rows_loaded = 4294960000;
district_rows_loaded = 0;

District();
Stock();

}

//=====
//
// Function : District
//
//=====
void District()
{
    short    d_id;
             long    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;
    long      w_id;
    RETCODE   rc;
    DBINT     rcint;
    char      bcphint[128];
    char      err_log_path[256];

    // Seed with unique number
    seed(4);

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

```

```

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

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");

// strcpy(err_log_path,aptr->log_path);
// strcat(err_log_path,"district.err");
sprintf(err_log_path,"%sdistrict%d.err",aptr->log_path,aptr-
>starting_warehouse);
rc = bcp_init(w_hdbc1, name, NULL, err_log_path, DB_IN);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    sprintf(bcphint, "tablock, 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, SQLINT4, 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 = (long)aptr->starting_warehouse; w_id <=
(long)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;
    long    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;

```

```

long    s_i_id;
DBINT    rc;
DBINT    rcint;
char    bcphint[128];
char    err_log_path[256];

// Seed with unique number
seed(3);

// if build index before load...
if ((aptr->build_index == 1) && (aptr->index_order == 1))
    BuildIndex("idxstkc1");

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

//rc = bcp_init(w_hdbc1, name, NULL, "logs\\stock.err",
DB_IN);
//
// strcpy(err_log_path,aptr->log_path);
// strcat(err_log_path,"stock.err");
// sprintf(err_log_path,"%sstock%d.err",aptr->log_path,aptr-
>starting_warehouse);
rc = bcp_init(w_hdbc1, name, NULL, err_log_path, DB_IN);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    sprintf(bcpint, "tablock, ROWS_PER_BATCH =
%u", (aptr->num_warehouses * 100000));
    rc = bcp_control(w_hdbc1, BCPHINTS, (void*)
bcpint);

    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);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 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 = (long)aptr->starting_warehouse;
s_w_id <= (long)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_Big(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;
    long
w_id;

    short                d_id;

    DWORD
dwThreadID[MAX_CUSTOMER_THREADS];
    HANDLE
hThread[MAX_CUSTOMER_THREADS];
    char                name[20];
    RETCODE
rc;
    DBINT
rcint;

```

```

char
bcphint[128];
char
cmd[256];
int
num_procs;

char
err_log_path_cust[256];
char
err_log_path_hist[256];

// 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("idxcustcl");
    // check the number of processors on this system
    // if 8 or more processors, then build index on
History.
    // if less than 8 processors, do not build the index
    num_procs = atoi(getenv(
"NUMBER_OF_PROCESSORS"));
    if ( num_procs >= 8 )
        BuildIndex("idxhistcl");
}

// Initialize bulk copy
sprintf(name, "%s.%s", aptr->database, "customer");

//rc = bcp_init(c_hdbc1, name, NULL, "logs\\customer.err",
DB_IN);
//
// strcpy(err_log_path_cust,aptr->log_path);
// strcat(err_log_path_cust,"customer.err");
// sprintf(err_log_path_cust,"%scustomer%d.err",aptr-
>log_path,aptr->starting_warehouse);
rc = bcp_init(c_hdbc1, name, NULL, err_log_path_cust,
DB_IN);

if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    sprintf(bcphint, "tablock, ROWS_PER_BATCH =
%u", (aptr->num_warehouses * 30000));
    rc = bcp_control(c_hdbc1, BCPHINTS, (void*)
bcphint);

    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc1);

```

```

    }

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

    // rc = bcp_init(c_hdbc2, name, NULL, "logs\\history.err", DB_IN);
    // strcpy(err_log_path_hist, aptr->log_path);
    // strcat(err_log_path_hist, "history.err");
    // sprintf(err_log_path_cust, "%shistory%d.err", aptr-
>log_path, aptr->starting_warehouse);
    rc = bcp_init(c_hdbc2, name, NULL, err_log_path_hist, DB_IN);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    sprintf(bcphint, "tablock");
    rc = bcp_control(c_hdbc2, BCPHINTS, (void*) bcphint);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    customer_rows_loaded = 0;
    history_rows_loaded = 0;

    CustomerBufInit();

    customer_time_start.time_start = (TimeNow() / MILLI);
    history_time_start.time_start = (TimeNow() / MILLI);

    for (w_id = (long)aptr->starting_warehouse; w_id <=
(long)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

            // Start customer table thread

            printf("...Loading customer table for:
d_id = %ld, w_id = %ld\n", d_id, w_id);

            hThread[0] = CreateThread(NULL,

(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 = %ld, w_id = %ld\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("idxcusc1");
    // check the number of processors on this system
    // if 8 or more processors, then build index on
    History.
    // if less than 8 processors, do not build the index
    num_procs = atoi(getenv(
"NUMBER_OF_PROCESSORS" ));
    if (num_procs >= 8)
        BuildIndex("idxhisc1");
}

// build non-clustered index
if (aptr->build_index == 1)
    BuildIndex("idxcusnc");

// Output the NURAND used for the loader into C_FIRST for
C_ID = 1,
// C_W_ID = 1, and C_D_ID = 1
//sprintf(cmd, "osql -S%s -U%s -P%s -d%s -e -Q\\\"update
customer set c_first = 'C_LOAD = %d' where c_id = 1 and c_w_id = 1 and
c_d_id = 1\\\" > logs\\nurand_load.log",
/* if (aptr->starting_warehouse == 1)
{
    sprintf(cmd, "osql -S%s -U%s -P%s -d%s -e -
Q\\\"update customer set c_first = 'C_LOAD = %d' where c_id = 1 and c_w_id
= 1 and c_d_id = 1\\\" > %snurand_load.log",

```

```

>server,
>user,
>password,
>database,

        LOADER_NURAND_C,

>log_path);

    }

*/
    system(cmd);

    SQLFreeStmt(c_hstmt1, SQL_DROP);
    SQLDisconnect(c_hdbc1);
    SQLFreeConnect(c_hdbc1);

    SQLFreeStmt(c_hstmt2, SQL_DROP);
    SQLDisconnect(c_hdbc2);
    SQLFreeConnect(c_hdbc2);

    return;
}

//=====
//
// Function : CustomerBufInit
//
//=====
void CustomerBufInit()
{
    long 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,"");
    }
}

```

aptr-
aptr-
aptr-
aptr-
aptr-
aptr-

```

        strcpy(customer_buf[i].c_street_1,"");
        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

        //        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, long 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

```

problem.

```

        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 = %ld, 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_buff[i].c_credit[0] = 'B';
    customer_buff[i].c_credit[1] = 'C';

    customer_buff[i].c_credit_lim = 50000.0;

    customer_buff[i].c_discount = ((float)
RandomNumber(0L, 5000L)) / 10000.0;

    // fix to avoid ODBC float to numeric conversion
problem.
    // customer_buff[i].c_balance = -10.0;
    strcpy(customer_buff[i].c_balance, "-10.0");

    MakeAlphaString(300, 500, C_DATA_LEN,
customer_buff[i].c_data);

    // Generate HISTORY data
    MakeAlphaString(12, 24, H_DATA_LEN,
customer_buff[i].h_data);
}

//=====
//
// Function : LoadCustomerTable
//
//=====
void LoadCustomerTable(LOADER_TIME_STRUCT *customer_time_start)
{
    long i;
    long c_id;
    short c_d_id;
    long 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, SQLINT4, 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

```

```

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)
{
    long c_id;
    short c_d_id;
    long 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, SQLINT4, 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, SQLINT4, 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;
    long
short      w_id;
           d_id;
    DWORD
dwThreadId[MAX_ORDER_THREADS];
    HANDLE
hThread[MAX_ORDER_THREADS];
    char      name[20];
    RETCODE
rc;
    char
bcphint[128];
    char
err_log_path_ord[256];
    char
err_log_path_nord[256];
    char
err_log_path_ordl[256];

    // seed with unique number
    seed(6);

    printf("Loading orders...\n");

    // if build index before load...
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        BuildIndex("idxordcl");
        BuildIndex("idxnodcl");
        BuildIndex("idxodcl");
    }

    // initialize bulk copy
    sprintf(name, "%s..%s", aptr->database, "orders");
    // strcpy(err_log_path_ord, aptr->log_path);
    // strcat(err_log_path_ord, "orders.err");
    // sprintf(err_log_path_ord, "%sorders%d.err", aptr-
>log_path, aptr->starting_warehouse);
    rc = bcp_init(o_hdbc1, name, NULL, err_log_path_ord, DB_IN);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcphint, "tablock, ROWS_PER_BATCH =
%u", (aptr->num_warehouses * 30000));
        rc = bcp_control(o_hdbc1, BCPHINTS, (void*)
bcphint);
        if (rc != SUCCEED)

```

```

    }
        HandleErrorDBC(o_hdbc1);

    sprintf(name, "%s..%s", aptr->database, "new_order");
    // rc = bcp_init(o_hdbc2, name, NULL, "logs\\neword.err",
DB_IN);
    // strcpy(err_log_path_nord, aptr->log_path);
    // strcat(err_log_path_nord, "neword.err");
    // sprintf(err_log_path_ord, "%sneword%d.err", aptr-
>log_path, aptr->starting_warehouse);
    rc = bcp_init(o_hdbc2, name, NULL, err_log_path_nord,
DB_IN);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcphint, "tablock, ROWS_PER_BATCH =
%u", (aptr->num_warehouses * 9000));
        rc = bcp_control(o_hdbc2, BCPHINTS, (void*)
bcphint);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc2);
    }

    sprintf(name, "%s..%s", aptr->database, "order_line");
    // rc = bcp_init(o_hdbc3, name, NULL, "logs\\ordline.err",
DB_IN);
    // strcpy(err_log_path_ordl, aptr->log_path);
    // strcat(err_log_path_ordl, "ordline.err");
    // sprintf(err_log_path_ord, "%sordline%d.err", aptr-
>log_path, aptr->starting_warehouse);
    rc = bcp_init(o_hdbc3, name, NULL, err_log_path_ordl,
DB_IN);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcphint, "tablock, 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;
    order_line_rows_loaded = 4294960000;

```

```

    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 = (long)aptr->starting_warehouse; w_id <=
(long)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

            // start Orders table thread

            printf("...Loading Order Table for:
d_id = %ld, w_id = %ld\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 = %ld, w_id = %ld\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. LastError = %d\n", GetLastError());
        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(short d_id, long w_id)
{
    int cust[ORDERS_PER_DISTRICT+1];
    long o_id;
    long ol;

    printf("...Loading Order Buffer for: d_id = %ld, w_id = %ld\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)
        {

```

```

        (short)RandomNumber(1L, 10L);
        }
        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;
    long 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, SQLINT4, 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("idxordc1");

        // build non-clustered index
        if (aptr->build_index == 1)
            BuildIndex("idxordnc");
    }

}

//=====
//
// Function : LoadNewOrderTable
//
//=====

void LoadNewOrderTable(LOADER_TIME_STRUCT *new_order_time_start)
{
    long          long          i;
    short         o_id;
               o_d_id;
    long          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, SQLINT4, 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_Big(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("idxnodc1");
    }

}

//=====
//
// Function : LoadOrderLineTable
//
//=====

void LoadOrderLineTable(LOADER_TIME_STRUCT *order_line_time_start)
{

```

```

    long          long          j;
    long          o_id;
    short         o_d_id;
    long          o_w_id;
    double        ol;
    long          ol_i_id;
    long          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, SQLINT4, 3);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 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, SQLINT4, 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 =
                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_Big(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("idxodcl");
        }
    }

    //=====
    //
    // 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,
                                long
                                rows_loaded,
                                char
                                *table_name,
                                long *time_start)
    {
        long time_end, time_diff;

        if ( !(rows_loaded % aptr->batch) )
        {
            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 : CheckForCommit_Big
    //
    //=====
    void CheckForCommit_Big(HDBC hdbc,
                                HSTMT
                                hstmt,
                                double
                                rows_loaded,
                                char
                                *table_name,
                                long *time_start)

```

```

{
    long    time_end, time_diff;

    if ( !(fmod(rows_loaded,aptr->batch) ) )
    {
        time_end = (TimeNow() / MILLI);
        time_diff = time_end - *time_start;

        printf("-> Loaded %ld rows into %s in %ld sec -
Total = %.0f (%.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;
    int         retrycnt;
    int         done;

    SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE,
&henv);

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

```

```

// Open connections to SQL Server

// Connection 1

if (aptr->tables_all || aptr->table_item)
{
    retrycnt = 1;
    done = 0;

    SQLAllocHandle(SQL_HANDLE_DBC, henv ,
&i_hdbc1);

    SQLSetConnectAttr(i_hdbc1, SQL_COPT_SS_BCP,
(void *)SQL_BCP_ON, SQL_IS_INTEGER );
    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);

    while (!done)
    {
        rc = SQLDriverConnect ( i_hdbc1,

            NULL,

            (SQLCHAR*)&szDriverString[0] ,

            SQL_NTS,

            (SQLCHAR*)&szDriverStringOut[0],

            sizeof(szDriverStringOut),

            &cbDriverStringOut,

            SQL_DRIVER_NOPROMPT );
        if ( (rc != SUCCEED) &&
            (rc !=

                SQL_SUCCESS_WITH_INFO) )

            {
                if (retrycnt <= aptr-
>retry_limit)
                    {

```

```

            retrycnt++;
            Sleep(aptr->sleep_milli);
        } else
        {
            HandleErrorDBC(i_hdbc1);

            printf("TPC-C Loader aborted!\n");
            exit(9);
        }
    } else
        done = 1;
}

} // of if item or all

// Connection 2

if (aptr->tables_all || aptr->table_warehouse)
{
    retrycnt = 1;
    done = 0;

    SQLAllocHandle(SQL_HANDLE_DBC, henv ,
&w_hdbc1);

    SQLSetConnectAttr(w_hdbc1, SQL_COPT_SS_BCP,
(void *)SQL_BCP_ON, SQL_IS_INTEGER );
    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);

    while (!done)
    {
        rc = SQLDriverConnect ( w_hdbc1,

            NULL,

            (SQLCHAR*)&szDriverString[0] ,

```

```

        SQL_NTS,
        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
    if ( (rc != SUCCEED) &&
        (rc !=
SQL_SUCCESS_WITH_INFO) )
    {
        if (retrycnt <= aptr-
>retry_limit)
        {
            retrycnt++;
            Sleep(aptr->sleep_milli);
        } else
        {
            HandleErrorDBC(w_hdbc1);
            printf("TPC-C Loader aborted!\n");
            exit(9);
        }
    } else
    {
        done = 1;
    }
} // of if all or warehouse

// Connection 3
if (aptr->tables_all || aptr->table_customer)
{
    retrycnt = 1;
    done = 0;

    SQLAllocHandle(SQL_HANDLE_DBC, henv ,
&c_hdbc1);
    SQLSetConnectAttr(c_hdbc1, SQL_COPT_SS_BCP,
(void *)SQL_BCP_ON, SQL_IS_INTEGER );
    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);

    while (!done)
    {
        rc = SQLDriverConnect ( c_hdbc1,
        NULL,
        (SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,
        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
        if ( (rc != SUCCEED) &&
            (rc !=
SQL_SUCCESS_WITH_INFO) )
        {
            if (retrycnt <= aptr-
>retry_limit)
            {
                retrycnt++;
                Sleep(aptr->sleep_milli);
            } else
            {
                HandleErrorDBC(c_hdbc1);
                printf("TPC-C Loader aborted!\n");
                exit(9);
            }
        } else
        {
            done = 1;
        }
    }
}

```

```

// Connection 4
retrycnt = 1;
done = 0;

SQLAllocHandle(SQL_HANDLE_DBC, henv ,
&c_hdbc2);
    SQLSetConnectAttr(c_hdbc2, SQL_COPT_SS_BCP,
(void *)SQL_BCP_ON, SQL_IS_INTEGER );
    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);

    while (!done)
    {
        rc = SQLDriverConnect ( c_hdbc2,
        NULL,
        (SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,
        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
        if ( (rc != SUCCEED) &&
            (rc !=
SQL_SUCCESS_WITH_INFO) )
        {
            if (retrycnt <= aptr-
>retry_limit)
            {
                retrycnt++;
                Sleep(aptr->sleep_milli);
            } else

```

```

        {
            HandleErrorDBC(o_hdbc2);

            printf("TPC-C Loader aborted!\n");

            exit(9);
        }
        else
            done = 1;
    }

    // of if all or customer
    if (aptr->tables_all || aptr->table_orders)
    {
        retrycnt = 1;
        done = 0;

        SQLAllocHandle(SQL_HANDLE_DBC, henv,
            &o_hdbc1);
        SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP,
            (void *)SQL_BCP_ON, SQL_IS_INTEGER);
        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);

        while (!done)
        {
            rc = SQLDriverConnect ( o_hdbc1,

                NULL,

                (SQLCHAR*)&szDriverString[0],

                SQL_NTS,

                (SQLCHAR*)&szDriverStringOut[0],

                sizeof(szDriverStringOut),

```

```

                &cbDriverStringOut,
                SQL_DRIVER_NOPROMPT );
                if ( (rc != SUCCEED) &&
                    (rc !=
                        SQL_SUCCESS_WITH_INFO) )
                {
                    if (retrycnt <= aptr-
                        >retry_limit)
                    {
                        retrycnt++;
                        Sleep(aptr->sleep_milli);
                    }
                    else
                    {
                        HandleErrorDBC(o_hdbc1);
                        printf("TPC-C Loader aborted!\n");
                        exit(9);
                    }
                }
                else
                    done = 1;
            }

            // Connection 6

            retrycnt = 1;
            done = 0;

            SQLAllocHandle(SQL_HANDLE_DBC, henv,
                &o_hdbc2);
            SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP,
                (void *)SQL_BCP_ON, SQL_IS_INTEGER );
            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);

            while (!done)

```

```

        {
            rc = SQLDriverConnect ( o_hdbc2,

                NULL,

                (SQLCHAR*)&szDriverString[0],

                SQL_NTS,

                (SQLCHAR*)&szDriverStringOut[0],

                sizeof(szDriverStringOut),

                &cbDriverStringOut,

                SQL_DRIVER_NOPROMPT );
                if ( (rc != SUCCEED) &&
                    (rc !=
                        SQL_SUCCESS_WITH_INFO) )
                {
                    if (retrycnt <= aptr-
                        >retry_limit)
                    {
                        retrycnt++;
                        Sleep(aptr->sleep_milli);
                    }
                    else
                    {
                        HandleErrorDBC(o_hdbc2);
                        printf("TPC-C Loader aborted!\n");
                        exit(9);
                    }
                }
                else
                    done = 1;
            }

            // Connection 7

            retrycnt = 1;
            done = 0;

            SQLAllocHandle(SQL_HANDLE_DBC, henv,
                &o_hdbc3);
            SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP,
                (void *)SQL_BCP_ON, SQL_IS_INTEGER );
            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);

    while (!done)
    {
        rc = SQLDriverConnect ( o_hdbc3,

        NULL,

        (SQLCHAR*)&szDriverString[0] ,

        SQL_NTS,

        (SQLCHAR*)&szDriverStringOut[0],

        sizeof(szDriverStringOut),

        &cbDriverStringOut,

        SQL_DRIVER_NOPROMPT );
        if ( (rc != SUCCEED) &&
            (rc !=
SQL_SUCCESS_WITH_INFO) )
        {
            if (retrycnt <= aptr-
>retry_limit)

                retrycnt++;

                Sleep(aptr->sleep_milli);

            } else
            {
                HandleErrorDBC(o_hdbc3);

                printf("TPC-C Loader aborted!\n");

                exit(9);

            } else
                done = 1;

        }

    } //
}

```

```

//=====
//
// Function name: BuildIndex
//
//=====
void BuildIndex(char *index_script)
{
    char cmd[256];

    printf("Starting index creation: %s\n",index_script);

    sprintf(cmd, "osql -S%s -U%s -P%s -e -i%s\\%s.sql >
%s%s.log",

        aptr->server,
        aptr->user,
        aptr->password,
        aptr-
        >index_script_path,

        index_script,
        aptr->log_path,
        index_script);

    system(cmd);

    printf("Finished index creation: %s\n",index_script);
}

//=====
//
// Function name: HandleErrorDBC
//
//=====
void HandleErrorDBC (SQLHDBC hdbc1)
{
    SQLCHAR SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
SQLINTEGER NativeError;
SQLSMALLINT i, MsgLen;
SQLRETURN rc2;
    char timebuf[128];
    char datebuf[128];
    char err_log_path[256];
    FILE *fp1;

    i = 1;

```

```

        while (( rc2 = SQLGetDiagRec(SQL_HANDLE_DBC , hdbc1, i,
SqlState , &NativeError,
        &MsgLen )) != SQL_NO_DATA )
        {
            if (totaldberrors >= MAXSQLERRORS)
            {
                printf(">>>>>Max SQL errors of
%d exceeded, terminating tpccldr<<<<<<\n",totaldberrors);
                exit(9);
            }
            sprintf( szLastError, "%s" , Msg );

            _strtime(timebuf);
            _strdate(datebuf);

            printf( "[%s : %s] %s\nSQLState: %s\n" ,
datebuf, timebuf, szLastError, SqlState);

            strcpy(err_log_path,aptr->log_path);
            strcat(err_log_path,"tpccldr.err");
            sprintf(err_log_path,"%stpcldr%d.err",aptr->log_path,aptr-
>starting_warehouse);

            fp1 = fopen(err_log_path,"a+");
            if (fp1 == NULL)
                printf("ERROR: Unable to open
errorlog file.\n");

            else
            {
                fprintf(fp1, "[%s : %s]
%s\nSQLState: %s\n" , datebuf, timebuf, szLastError, SqlState);
                fclose(fp1);
            }

            i++;
        }

    }

//=====
//
// Function : HandleErrorSTMT
//
//=====
void HandleErrorSTMT (HSTMT hstmt1)
{
    SQLCHAR SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
SQLINTEGER NativeError;

```

```

SQLSMALLINT i, MsgLen;
SQLRETURN rc2;
char          timebuf[128];
char          datebuf[128];
char          err_log_path[256];
FILE          *fp1;

i = 1;
while (( rc2 = SQLGetDiagRec(SQL_HANDLE_STMT , hstmt1, i,
SqlState , &NativeError,
                                Msg, sizeof(Msg) ,
                                &MsgLen )) != SQL_NO_DATA )
{
    if (totaldberrors >= MAXSQLERRORS)
    {
        printf(">>>>>Max SQL errors of
%d exceeded, terminating tpccldr<<<<<<\n",totaldberrors);
        exit(9);
    }
    totaldberrors++;

    sprintf( szLastError , "%s" , Msg );

    _strtime(timebuf);
    _strdate(datebuf);

    printf( "[%s : %s] %s\nSQLState: %s\n" ,
datebuf, timebuf, szLastError, SqlState);

    //      strcpy(err_log_path,aptr->log_path);
    //      strcat(err_log_path,"tpccldr.err");
    sprintf(err_log_path,"%stpcldr%d.err",aptr->log_path,aptr-
>starting_warehouse);
    fp1 = fopen(err_log_path,"a+");
    if (fp1 == NULL)
        printf("ERROR: Unable to open
errorlog file.\n");
    else
    {
        fprintf(fp1, "[%s : %s]
%s\nSQLState: %s\n" , datebuf, timebuf, szLastError, SqlState);
        fclose(fp1);
    }

    i++;
}

//=====
//

```

```

// Function : FormatDate
//=====
void FormatDate ( char* szTimeCOutput )
{
    struct tm when;
    time_t now;

    time( &now );
    when = *localtime( &now );

    mktime( &when );

    // odbc datetime format
    strftime( szTimeCOutput , 30 , "%Y-%m-%d %H:%M:%S.000" ,
&when );

    return;
}

//=====
//
// Function : CheckDataBase
//
//=====
void CheckDataBase()
{
    RETCODE rc;

    char          szDriverString[300];
    char          szDriverStringOut[1024];
    char          TablesBitMap[9] =
{"0000000000"};
    int          i,
ExitFlag;

    SQLSMALLINT          cbDriverStringOut;
    SQLCHAR              TabName[10];

#ifdef _WIN64
    SQLINTEGER          TabCount;
    SQLLEN              TabNameInd,
TabCountInd;
#else
    SQLINTEGER          TabNameInd, TabCount,
TabCountInd;

```

```

#endif
    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_INTEGER );
    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");
    }

```

```

creation errors.\n");

        printf("\nCheck LOGS\\ directory for database

        // 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
        exit(1);

SETUP.CMD
    }

    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);

```

```

each found table

SQL_NO_DATA)

    // go through the result set and set the bitmap for
    // set the bitmap to '1' if the table name is found

    while ((rc = SQLFetch(v_hstmt)) !=

    {
        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;

        // iterate 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
                                }
                                break;
                                case 7:
                                if (TablesBitMap[i] ==
'0')                                {
                                printf("The Item table is missing or damaged.\n");
                                ExitFlag
                                }
                                break;
                                case 8:
                                if (TablesBitMap[i] ==
'0')                                {
                                printf("The Stock table is missing or damaged.\n");
                                ExitFlag
                                }
                                break;
                                }
                                // if one or more tables are missing, display
message and exit the loader
                                if (ExitFlag = 1)
                                {
                                printf("\nExiting TPC-C Loader!\n");
                                printf("\nCheck LOGS\\ directory for
database\n");
                                printf("or table creation errors.\n");

                                // cleanup database connections and
handles
                                SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
                                SQLDisconnect(v_hdbc);

```

```

                                SQLFreeHandle(SQL_HANDLE_DBC,
v_hdbc);
                                exit(1);
                                }
                                // cleanup database connections and handles
                                SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
                                SQLDisconnect(v_hdbc);
                                SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);
                                return;
                                }

```

time.c

```

//      File:                TIME.C
//      Ver. 4.42
//      1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
//      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;
}

```

strings.c

```

//      File:                STRINGS.C
//      Ver. 4.42
//      1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
//      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 percentage is valid
    if ((percent < 0) || (percent > 100))
    {
        printf("MakeOriginalAlphaString: 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("[%d]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("[%d]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;
}

random.c

// File: RANDOM.C
// Microsoft TPC-C Kit
Ver. 4.42
// Copyright Microsoft,
1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
// 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("[%d]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("[%d]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("[%d]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("[%d]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("[%d]DBG: RandomNumber between %ld & %ld ==> %ld\n",
(int)
GetCurrentThreadId(), lower, upper, rand_num);
#endif

return rand_num;
}

#ifdef 0
//Original code pgd 08/13/96
```

```

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

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

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

#ifdef DEBUG

```

```

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

    return rand_num;
}

```

getargs.c

```

//      File:          GETARGS.C
//
//      Ver. 4.42
//
//      Copyright Microsoft,
//      1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
//      Purpose:      Source file for command line processing

// Includes
#include "tpcc.h"

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

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

#ifdef DEBUG
    printf("[%d]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_orders    =

FALSE;
pargs->loader_res_file = LOADER_RES_FILE;
pargs->log_path        = LOADER_LOG_PATH;
pargs->pack_size       = DEFLDPACKSIZE;
pargs->starting_warehouse =

DEF_STARTING_WAREHOUSE;
pargs->build_index     =

BUILD_INDEX;
pargs->index_order     =

INDEX_ORDER;
pargs->index_script_path = INDEX_SCRIPT_PATH;
pargs->scale_down      =

SCALE_DOWN;
pargs->sleep_milli     =

SLEEPMILLI;
pargs->retry_limit     =

RETRYLIMIT;

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

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

    ptr = argv[i];

    switch (ptr[1])
    {
        case '?': /* Fall through */
            GetArgsLoaderUsage();
            break;

        case 'D':
            pargs->database =

ptr+2;
            break;

        case 'P':
            pargs->password =

ptr+2;
            break;

        case 'S':

```

140

```

        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);
}

```

B.1 Database Options

DBOPT1 . SQL

```

-- File:   DBOPT1.SQL
--         Microsoft TPC-C Benchmark Kit Ver. 4.41
--         Copyright Microsoft, 2001
-- Purpose: Sets database options for data load

use master
go

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

use tpcc
go

checkpoint

```

go

DBOPT2 . SQL

```

-- File:   DBOPT2.SQL
--         Microsoft TPC-C Benchmark Kit Ver. 4.41
--         Copyright Microsoft, 2001
-- Purpose: Resets database options after data load

exec sp_dboption tpcc,'select into/bulkcopy',false
exec sp_dboption tpcc,'trunc. log on chkpt.',false
exec sp_dboption tpcc,'torn page detection',false
GO

USE tpcc
GO

CHECKPOINT
GO

sp_configure 'allow updates',1
GO

RECONFIGURE WITH OVERRIDE
GO

DECLARE    @msg                      varchar(50)

--
--
-- OPTIONS FOR SQL SERVER 2000
-- Set option values for user-defined indexes --
--
--

SET        @msg                      = ' '
PRINT      @msg

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

SET        @msg                      = ' '
PRINT      @msg

EXEC sp_indexoption 'customer', 'DisallowPageLocks', TRUE
EXEC sp_indexoption 'district', 'DisallowPageLocks', TRUE
EXEC sp_indexoption 'warehouse', 'DisallowPageLocks', TRUE
EXEC sp_indexoption 'stock', 'DisallowPageLocks', TRUE
EXEC sp_indexoption 'order_line', 'DisallowRowLocks', TRUE
EXEC sp_indexoption 'orders', 'DisallowRowLocks', TRUE
EXEC sp_indexoption 'new_order', 'DisallowRowLocks', TRUE
EXEC sp_indexoption 'item', 'DisallowRowLocks', TRUE

```

```

EXEC sp_indexoption 'item', 'DisallowPageLocks',
GO TRUE

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

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

ORDER BY lockflags asc
GO

sp_configure 'allow updates',0
GO

RECONFIGURE WITH OVERRIDE
GO

EXEC sp_dboption tpcc, 'auto update statistics', FALSE
EXEC sp_dboption tpcc, 'auto create statistics', FALSE
GO

EXEC sp_tableoption 'district', 'pintable',true
EXEC sp_tableoption 'warehouse', 'pintable',true
EXEC sp_tableoption 'new_order', 'pintable',true
EXEC sp_tableoption 'item', 'pintable',true
GO

```

B.2 Table definitions

createtables.sql

```
--
-- HPTPCC Setup Kit
-- Drop and Create Tables
-- Tables are initially created in the load_fg filegroup
--
```

```
SET ANSI_NULL_DFLT_OFF ON
go
```

```
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 int,
    w_name char(10),
    w_street_1 char(20),
    w_street_2 char(20),
```

```
    w_state char(20),
    w_zip char(9),
    w_tax numeric(4,4),
    w_ytd numeric(12,2)
```

```
) on load_fg
go
```

```
create table district
```

```
(
    d_id tinyint,
    d_w_id int,
    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 load_fg
go
```

```
create table customer
```

```
(
    c_id int,
    c_d_id tinyint,
    c_w_id int,
    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 smallint,
    c_payment_cnt smallint,
    c_delivery_cnt char(500),
    c_data
```

```
) on load_fg
go
```

```
char(20),
char(9),
```

```
create table history
```

```
(
    h_c_id int,
    h_c_d_id tinyint,
    h_c_w_id int,
    h_d_id tinyint,
    h_w_id int,
    h_date datetime,
    h_amount numeric(6,2),
    h_data char(24)
```

```
) on load_fg
go
```

```
create table new_order
```

```
(
    no_o_id int,
    no_d_id tinyint,
    no_w_id int
```

```
) on load_fg
go
```

```
create table orders
```

```
(
    o_id int,
    o_d_id tinyint,
    o_w_id int,
    o_c_id int,
    o_entry_d datetime,
    o_carrier_id tinyint,
    o_ol_cnt tinyint,
    o_all_local tinyint
```

```
) on load_fg
go
```

```
create table order_line
```

```
(
    ol_o_id int,
    ol_d_id tinyint,
    ol_w_id int,
    ol_number tinyint,
    ol_i_id int,
    ol_supply_w_id int,
    ol_delivery_d datetime,
    ol_quantity smallint,
    ol_amount numeric(6,2),
    ol_dist_info char(24)
```

```
) on load_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 load_fg
    go

create table stock
(
    s_i_id                    int,
    s_w_id                    int,
    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 load_fg
go

sp_tableoption 'warehouse','table lock on bulk load','on'
go
sp_tableoption 'district','table lock on bulk load','on'
go
sp_tableoption 'customer','table lock on bulk load','on'
go
sp_tableoption 'history','table lock on bulk load','on'
go
sp_tableoption 'new_order','table lock on bulk load','on'
go
sp_tableoption 'orders','table lock on bulk load','on'
go
sp_tableoption 'order_line','table lock on bulk load','on'
go
sp_tableoption 'stock','table lock on bulk load','on'
go

```

B.3 Stored Procedures

createnewordproc.sql

```

-- File: NEWORD.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.42
-- Copyright Microsoft, 2003
--
-- - 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 int,
    @d_id tinyint,
    @c_id int,
    @o_no int,
    @o_id int,
    @o_qty tinyint,
    @o_all_local tinyint,
    @i_id1 int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
    @i_id2 int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
    @i_id3 int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
    @i_id4 int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
    @i_id5 int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
    @i_id6 int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,

```

```

    @i_id7 int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
    @i_id8 int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
    @i_id9 int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
    @i_id15 int = 0, @s_w_id15 int = 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 int,
    @li_qty smallint,
    @ol_number int,
    @c_id_local int

begin

begin transaction n

-- get district tax and next available order id and update
-- plus initialize local variables

update
set
    district
    @d_tax = d_tax,

```



```

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,

@l_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 WITH (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 WITH (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

if exists ( select name from sysobjects where name = 'sp0tpcc_neworder' )
drop procedure sp0tpcc_neworder

go

create proc sp0tpcc_neworder

                                @w_id
int,
                                @d_id
tinyint,
                                @c_id
int,

                                @o_ol_cnt tinyint,

                                @o_all_local tinyint,

                                @i_id1
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
                                @i_id2
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
                                @i_id3
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
                                @i_id4
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
                                @i_id5
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
                                @i_id6
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
                                @i_id7
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
                                @i_id8
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
                                @i_id9
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
                                @i_id10
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
                                @i_id11
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
                                @i_id12
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,

```

```

int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 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  int,
           @li_qty     smallint,
           @ol_number  int,
           @c_id_local int

begin
begin transaction n

-----
-- get district tax and next available 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

@i_id13 -----while (@li_no < @o_ol_cnt)
@i_id14 begin
@i_id15         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
                        when 2 then
                        when 3 then
                        when 4 then
                        when 5 then
                        when 6 then
                        when 7 then
                        when 8 then
                        when 9 then
                        when 10 then
                        when 11 then
                        when 12 then
                        when 13 then
                        when 14 then
                        when 15 then
                        end,
                  @s_w_id1
                  @s_w_id2
                  @s_w_id3
                  @s_w_id4
                  @s_w_id5
                  @s_w_id6
                  @s_w_id7
                  @s_w_id8
                  @s_w_id9
                  @s_w_id10
                  @s_w_id11
                  @s_w_id12

                  @li_qty = case @li_no
                        when 1 then
                        when 2 then
                        when 3 then
                        when 4 then
                        when 5 then
                        when 6 then
                        when 7 then
                        when 8 then
                        when 9 then
                        when 10 then
                        when 11 then
                        when 12 then
                        when 13 then
                        when 14 then
                        when 15 then
                        end,
                  @ol_qty1
                  @ol_qty2
                  @ol_qty3
                  @ol_qty4
                  @ol_qty5
                  @ol_qty6
                  @ol_qty7
                  @ol_qty8
                  @ol_qty9
                  @ol_qty10
                  @ol_qty11
                  @ol_qty12
                  @ol_qty13
                  @ol_qty14
                  @ol_qty15

                  -- get item data (no one updates item)
                  -----
                  select    @i_price = i_price,
                            @i_name  = i_name,
                            @i_data  = i_data
                  from       item WITH (tablock, repeatableread)
                  where     i_id = @li_id

                  -----
                  -- update stock values
                  -----
                  update    stock

```

```

set s_ytd = s_ytd
+ @li_qty,
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 WITH (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 WITH (repeatableread)
where w_id = @w_id
if (@commit_flag = 1)
commit transaction n
else
-----
-- all that work for nuthn!!!
-----
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

if exists ( select name from sysobjects where name = 'sp1tpcc_neworder' )
    drop procedure sp1tpcc_neworder

go

create proc sp1tpcc_neworder

    @w_id          int,
    @d_id          tinyint,
    @c_id          int,
    @o_cnt         tinyint,
    @o_all_local   tinyint,
    @s_w_id1 int = 0, @s_w_id2 int = 0, @ol_qty1 smallint = 0,
    @s_w_id3 int = 0, @ol_qty2 smallint = 0,
    @s_w_id4 int = 0, @ol_qty3 smallint = 0,
    @s_w_id5 int = 0, @ol_qty4 smallint = 0,
    @s_w_id6 int = 0, @ol_qty5 smallint = 0,
    @s_w_id7 int = 0, @ol_qty6 smallint = 0,
    @s_w_id8 int = 0, @ol_qty7 smallint = 0,
    @s_w_id9 int = 0, @ol_qty8 smallint = 0,
    @s_w_id10 int = 0, @ol_qty9 smallint = 0,
    @s_w_id11 int = 0, @ol_qty10 smallint = 0,
    @s_w_id12 int = 0, @ol_qty11 smallint = 0,
    @s_w_id13 int = 0, @ol_qty12 smallint = 0,
    @s_w_id14 int = 0, @ol_qty13 smallint = 0,
    @s_w_id15 int = 0, @ol_qty14 smallint = 0,
    @s_w_id15 int = 0, @ol_qty15 smallint = 0

as

```

```

declare @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 int,
        @li_qty smallint,
        @ol_number int,
        @c_id_local int

begin

begin transaction n

-----
-- get district tax and next available 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
                                                when 2 then
                                                when 3 then
                                                when 4 then
                                                when 5 then
                                                when 6 then
                                                when 7 then
                                                when 8 then
                                                when 9 then
                                                when 10 then
                                                when 11 then
                                                when 12 then
                                                when 13 then
                                                when 14 then
                                                when 15 then
                                                end,
                                @li_qty = case @li_no
                                                when 1 then
                                                when 2 then

```

```

@ol_qty3          when 3 then
@ol_qty4          when 4 then
@ol_qty5          when 5 then
@ol_qty6          when 6 then
@ol_qty7          when 7 then
@ol_qty8          when 8 then
@ol_qty9          when 9 then
@ol_qty10         when 10 then
@ol_qty11         when 11 then
@ol_qty12         when 12 then
@ol_qty13         when 13 then
@ol_qty14         when 14 then
@ol_qty15         when 15 then
end

-----
-- get item data (no one updates item)
-----
select      @i_price = i_price,
            @i_name  = i_name,
            @i_data  = i_data
from        item WITH (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,

when 3 then
when 4 then
when 5 then
when 6 then
when 7 then
when 8 then
when 9 then
when 10 then
when 11 then
when 12 then
when 13 then
when 14 then
when 15 then
end

-----
-- 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,
@li_s_w_id,
@w_id,
@li_no,
@li_id,
@li_s_w_id,
'dec 31, 1899',
@li_qty,
@i_price * @li_qty,
@s_dist
= case
when 1
when 2
when 3
when 4
when 5
when 6
when 7
when 8
when 9
when
end
= @li_id
=
select ",0",0,0
select @commit_flag = 0

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 WITH (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,
@li_s_w_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 WITH (repeatableread)
where         w_id        = @w_id

if (@commit_flag = 1)

        commit transaction n
else

-----
-- all that work for nuthn!!!
-----
        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
if exists ( select name from sysobjects where name = 'sp2tpcc_neworder' )
        drop procedure sp2tpcc_neworder

go

create proc sp2tpcc_neworder

int,

```

```

tinyint,
int,

        @o_ol_cnt      tinyint,

        @o_all_local   tinyint,

int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 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),

```

```

@d_id
@c_id

@i_id1
@i_id2
@i_id3
@i_id4
@i_id5
@i_id6
@i_id7
@i_id8
@i_id9
@i_id10
@i_id11
@i_id12
@i_id13
@i_id14
@i_id15

```

```

@s_dist@li_nochar(24)nt,
        @o_id      int,
        @commit_flag tinyint,
        @li_id     int,
        @li_s_w_id int,
        @li_qty     smallint,
        @ol_number int,
        @c_id_local int

begin

begin transaction n

-----
-- get district tax and next available 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 WITH (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
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) )

```

```

end,
                                then 'B' else 'G'
                                @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 WITH (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,

```

```

-----
-- select warehouse tax
-----
select      @w_tax   = w_tax
from        warehouse WITH (repeatableread)
where      w_id      = @w_id

if (@commit_flag = 1)
    commit transaction n
else
    rollback transaction n
-----
-- all that work for nuthin!!!
-----
-- 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
if exists ( select name from sysobjects where name = 'sp3tpcc_neworder' )
    drop procedure sp3tpcc_neworder
go
create proc sp3tpcc_neworder
    @w_id
    int,
    @d_id
    tinyint,
    @c_id
    int,
    @o_ol_cnt    tinyint,
    @o_all_local tinyint,
    @i_id1
    int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
    @i_id2
    int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,

```

```

                                @i_id3
                                @i_id4
                                @i_id5
                                @i_id6
                                @i_id7
                                @i_id8
                                @i_id9
                                @i_id10
                                @i_id11
                                @i_id12
                                @i_id13
                                @i_id14
                                @i_id15

int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 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  int,
            @li_qty     smallint,
            @ol_number  int,
            @c_id_local int
begin

```



```

        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,

                @l_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 WITH (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 WITH (repeatableread)
        where      w_id          = @w_id

        if (@commit_flag = 1)

```

```

        else
        commit transaction n

-----
-- 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
if exists ( select name from sysobjects where name = 'sp4tpcc_neworder' )
    drop procedure sp4tpcc_neworder
go

create proc sp4tpcc_neworder

        @w_id
        int,
        @d_id
        tinyint,
        @c_id
        int,

        @o_ol_cnt    tinyint,
        @o_all_local tinyint,

        @i_id1
        int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
        @i_id2
        int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
        @i_id3
        int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
        @i_id4
        int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
        @i_id5
        int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
        @i_id6
        int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
        @i_id7
        int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
        @i_id8
        int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,

```

```

int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 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  int,
           @li_qty     smallint,
           @ol_number  int,
           @c_id_local int

begin

begin transaction n

-----
-- get district tax and next available 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,

           @li_id9
           @i_id10
           @i_id11
           @i_id12
           @i_id13
           @i_id14
           @i_id15

           @li_entry_d = @getdate(),
           @commit_flag = 1
           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
                           when 2 then
                           when 3 then
                           when 4 then
                           when 5 then
                           when 6 then
                           when 7 then
                           when 8 then

@s_w_id1
@s_w_id2
@s_w_id3
@s_w_id4
@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8

                           when 9 then
                           when 10 then
                           when 11 then
                           when 12 then
                           when 13 then
                           when 14 then
                           when 15 then

@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15

                           end,
                           @li_qty = case @li_no
                           when 1 then
                           when 2 then
                           when 3 then
                           when 4 then
                           when 5 then
                           when 6 then
                           when 7 then
                           when 8 then
                           when 9 then
                           when 10 then
                           when 11 then
                           when 12 then
                           when 13 then
                           when 14 then
                           when 15 then

@ol_qty1
@ol_qty2
@ol_qty3
@ol_qty4
@ol_qty5
@ol_qty6
@ol_qty7
@ol_qty8
@ol_qty9
@ol_qty10
@ol_qty11
@ol_qty12
@ol_qty13
@ol_qty14
@ol_qty15

                           end,
                           -- get item data (no one updates item)
                           -----
                           select    @i_price = i_price,
                                       @i_name = i_name,
                                       when 9 then
                                       when 10 then
                                       when 11 then
                                       when 12 then
                                       when 13 then
                                       when 14 then
                                       when 15 then
                                       end,
                                       @li_qty = case @li_no
                                       when 1 then
                                       when 2 then
                                       when 3 then
                                       when 4 then
                                       when 5 then
                                       when 6 then
                                       when 7 then
                                       when 8 then
                                       when 9 then
                                       when 10 then
                                       when 11 then
                                       when 12 then
                                       when 13 then
                                       when 14 then
                                       when 15 then
                                       end,
                                       @ol_qty1
                                       @ol_qty2
                                       @ol_qty3
                                       @ol_qty4
                                       @ol_qty5
                                       @ol_qty6
                                       @ol_qty7
                                       @ol_qty8
                                       @ol_qty9
                                       @ol_qty10
                                       @ol_qty11
                                       @ol_qty12
                                       @ol_qty13
                                       @ol_qty14
                                       @ol_qty15
                                       end,
                                       -- get item data (no one updates item)
                                       -----
                                       select    @i_price = i_price,
                                                   @i_name = i_name,

```

```

from      @i_data = i_data
where     item WITH (tablock, repeatableread)
         i_id = @li_id

-----
-- update stock values
-----

update    stock
set       s_ytd                = s_ytd

+ @li_qty,

@s_quantity = s_quantity =

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
10 then s_dist_10
when
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

-----
-- get customer last name, discount, and credit rating
-----

```

```

select    @c_discount = c_discount,
         @c_credit    = c_credit,
         @c_id_local  = c_id
from      customer WITH (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 WITH (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
if exists ( select name from sysobjects where name = 'sp5tpcc_neworder' )
    drop procedure sp5tpcc_neworder
go

create proc sp5tpcc_neworder
    @w_id          int,
    @d_id          tinyint,
    @c_id          int,
    @o_ol_cnt      tinyint,
    @o_all_local   tinyint,
    @i_id1         int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
    @i_id2         int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
    @i_id3         int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
    @i_id4         int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
    @i_id5         int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
    @i_id6         int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
    @i_id7         int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
    @i_id8         int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
    @i_id9         int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
    @i_id10        int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
    @i_id11        int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
    @i_id12        int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
    @i_id13        int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,

```

```

int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 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  int,
           @li_qty     smallint,
           @ol_number  int,
           @c_id_local int

begin

begin transaction n

-----
-- get district tax and next available 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

```

```

        @i_id14
        @i_id15

```

```

        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 WITH (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
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 WITH (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 WITH (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

```

```

drop procedure if exists sp6tpcc_neworder
go

create proc sp6tpcc_neworder
    @w_id int,
    @d_id tinyint,
    @c_id int,
    @o_ol_cnt tinyint,
    @o_all_local tinyint,
    @i_id1 int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
    @i_id2 int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
    @i_id3 int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
    @i_id4 int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
    @i_id5 int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
    @i_id6 int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
    @i_id7 int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
    @i_id8 int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
    @i_id9 int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
    @i_id15 int = 0, @s_w_id15 int = 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_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    int,
    @li_qty       smallint,
    @ol_number    int,
    @c_id_local   int

begin

begin transaction n

-----
-- get district tax and next available 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 4 then		when 2
	when 8 then @i_id8	@ol_qty4	when 5 then	then s_dist_02	when 3
	when 9 then @i_id9	@ol_qty5		then s_dist_03	
	when 10 then @i_id10		when 6 then		when 4
	when 11 then @i_id11	@ol_qty6		then s_dist_04	
	when 12 then @i_id12		when 7 then	then s_dist_05	when 5
	when 13 then @i_id13	@ol_qty7		then s_dist_06	when 6
	when 14 then @i_id14		when 8 then	then s_dist_07	when 7
	when 15 then @i_id15	@ol_qty8	when 9 then	then s_dist_08	when 8
	end,	@ol_qty9	when 10 then	then s_dist_09	when 9
@s_w_id1	@li_s_w_id = case @li_no	@ol_qty10	when 11 then	10 then s_dist_10	when
	when 1 then		when 12 then		
@s_w_id2	when 2 then	@ol_qty11	when 13 then		end
	when 3 then	@ol_qty12	when 14 then	and	= @li_id
@s_w_id3	when 4 then	@ol_qty13	when 15 then	@li_s_w_id	=
@s_w_id4	when 5 then	@ol_qty14	end		
@s_w_id5	when 6 then	@ol_qty15			
@s_w_id6	when 7 then				
@s_w_id7	when 8 then	-----		-----	
	when 9 then	-- get item data (no one updates item)		-- if there actually is a stock (and item) with these ids, go to work	
@s_w_id8	when 10 then	-----		if (@@rowcount > 0)	
@s_w_id9	when 11 then	select	@i_price = i_price,	begin	
	when 12 then		@i_name = i_name,		
@s_w_id10	when 13 then	from	@i_data = i_data	-----	
@s_w_id11	when 14 then	where	item WITH (tablock, repeatableread)	-- insert order_line data (using data from item and stock)	
@s_w_id12	when 15 then		i_id = @li_id	-----	
@s_w_id13	when 16 then	-----		insert into order_line values(@o_id,	
@s_w_id14	when 17 then	-- update stock values		@d_id,	
@s_w_id15	when 18 then	-----		@w_id,	
	when 19 then	update	stock	@li_no,	
	when 20 then	set	s_ytd	@li_id,	
	when 21 then	+ @li_qty,	@s_quantity = s_quantity	@li_s_w_id,	
	when 22 then	s_quantity - @li_qty +	= s_ytd	'dec 31, 1899',	
	when 23 then	case when (s_quantity - @li_qty < 10) then 91 else 0 end,		@li_qty,	
@ol_qty1	when 24 then	s_order_cnt = s_order_cnt + 1,		@i_price * @li_qty,	
@ol_qty2	when 25 then	s_remote_cnt =		@s_dist)	
@ol_qty3	when 26 then	s_remote_cnt + case when (@li_s_w_id = @w_id) then 0 else 1 end,		-----	
	when 27 then	@s_data			
	when 28 then	@s_dist			
	when 29 then	@d_id			
	when 30 then	then s_dist_01	when 1		

```

-- 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',0,0
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 WITH (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 WITH (repeatableread)
where       w_id        = @w_id

if (@commit_flag = 1)
            commit transaction n
else
            rollback transaction n

-----
-- all that work for nuthin!!!
-----
-- 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
if exists ( select name from sysobjects where name = 'sp7tpcc_neworder' )
drop procedure sp7tpcc_neworder

go
create proc sp7tpcc_neworder
int,
tinyint,
int,

```

```

            @o_ol_cnt    tinyint,
            @o_all_local tinyint,
            @i_id1
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
            @i_id2
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
            @i_id3
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
            @i_id4
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
            @i_id5
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
            @i_id6
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
            @i_id7
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
            @i_id8
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
            @i_id9
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
            @i_id10
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
            @i_id11
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
            @i_id12
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
            @i_id13
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
            @i_id14
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
            @i_id15
int = 0, @s_w_id15 int = 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  int,
@li_qty     smallint,
@ol_number  int,
@c_id_local int

begin

begin transaction n

-----
-- get district tax and next available 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

                                @ol_qty1
                                @ol_qty2
                                @ol_qty3
                                @ol_qty4
                                @ol_qty5
                                @ol_qty6
                                @ol_qty7
                                @ol_qty8
                                @ol_qty9

                                when 1 then
                                when 2 then
                                when 3 then
                                when 4 then
                                when 5 then
                                when 6 then
                                when 7 then
                                when 8 then
                                when 9 then

                                when 10 then
                                when 11 then
                                when 12 then
                                when 13 then
                                when 14 then
                                when 15 then

                                end,
                                @li_s_w_id = case @li_no
                                                when 1 then
                                                when 2 then
                                                when 3 then
                                                when 4 then
                                                when 5 then
                                                when 6 then
                                                when 7 then
                                                when 8 then
                                                when 9 then

                                                when 10 then
                                                when 11 then
                                                when 12 then
                                                when 13 then
                                                when 14 then
                                                when 15 then

                                                end,
                                                @ol_qty10
                                                @ol_qty11
                                                @ol_qty12
                                                @ol_qty13
                                                @ol_qty14
                                                @ol_qty15

                                                when 10 then
                                                when 11 then
                                                when 12 then
                                                when 13 then
                                                when 14 then
                                                when 15 then

                                                end

                                -----
                                -- get item data (no one updates item)
                                -----
                                select      @i_price = i_price,
                                             @i_name  = i_name,
                                             @i_data  = i_data
                                from        item WITH (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
                                when 1
                                when 2
                                when 3
                                when 4
                                when 5
                                when 6
                                when 7
                                when 10
                                when 11
                                when 12
                                when 13
                                when 14
                                when 15
                                end

                                @d_id

                                then s_dist_01
                                then s_dist_02
                                then s_dist_03
                                then s_dist_04
                                then s_dist_05
                                then s_dist_06
                                then s_dist_07

```

```

then s_dist_08
when 8
then s_dist_09
when 9
10 then s_dist_10
when
where s_i_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 * @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 WITH (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 WITH (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

```

createordstatproc.sql

```

-----
-- File: ORDSTAT.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.42
-- Copyright Microsoft, 2003
--
-- - 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      int,
                                @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 WITH (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 WITH (repeatableread)
        where       c_last    =
@c_last and

```

```

and
        order      c_w_id      = @w_id
                   c_d_id      = @d_id
                   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 WITH (repeatableread)
            where       c_id      = @c_id

            and

            where      c_d_id      = @d_id

            and

            where      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 WITH (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 WITH (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

if exists ( select name from sysobjects where name = 'sp0tpcc_orderstatus' )
    drop procedure sp0tpcc_orderstatus
go

create proc sp0tpcc_orderstatus      @w_id      int,
                                    @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 WITH (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 WITH (repeatableread)
where       c_last     =

@c_last and
and
            c_w_id      = @w_id

            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 WITH (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 WITH (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 WITH (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

```

```

if exists ( select name from sysobjects where name = 'sp1tpcc_orderstatus' )
drop procedure sp1tpcc_orderstatus
go

create proc sp1tpcc_orderstatus      @w_id      int,
                                     @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 WITH (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 WITH (repeatableread)
where       c_last     =

@c_last and
and
            c_w_id      = @w_id

            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 WITH (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 WITH (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_amount,
            ol_delivery_d
from        order_line WITH (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
if exists ( select name from sysobjects where name = 'sp2tpcc_orderstatus' )
    drop procedure  sp2tpcc_orderstatus
go

create proc sp2tpcc_orderstatus      @w_id      int,
                                     @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 WITH (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 WITH (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 WITH (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 WITH (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 WITH (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
if exists ( select name from sysobjects where name = 'sp3tpcc_orderstatus' )
    drop procedure  sp3tpcc_orderstatus
go

create proc sp3tpcc_orderstatus      @w_id      int,

```

```

            @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 WITH (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 WITH (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 WITH (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 WITH (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 WITH (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
if exists ( select name from sysobjects where name = 'sp4tpcc_orderstatus' )
    drop procedure  sp4tpcc_orderstatus
go

create proc sp4tpcc_orderstatus      @w_id      int,
                                     @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 WITH (repeatableread)
        where       c_last    = @c_last and

```

```

        c_d_id      = @d_id and
        c_w_id      = @w_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 WITH (repeatableread)
where     c_last     =

@c_last and

        c_w_id      = @w_id

and

        c_d_id      = @d_id
        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 WITH (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 WITH (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 WITH (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
if exists ( select name from sysobjects where name = 'sp5tpcc_orderstatus' )
    drop procedure  sp5tpcc_orderstatus
go

create proc sp5tpcc_orderstatus      @w_id      int,
                                     @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 WITH (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 WITH (repeatableread)
        where       c_last    =

@c_last and
        c_w_id      = @w_id

and
        c_d_id      = @d_id
        order by    c_w_id, c_d_id, c_last, c_first

        set          rowcount 0

    end

else
begin
-----
-- get customer info if by id
-----
        select      @c_balance = c_balance,

```

```

        @c_middle = c_middle,
        @c_last   = c_last
        from        customer WITH (repeatableread)
        where       c_id      = @c_id

        c_d_id      = @d_id

        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 WITH (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 WITH (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
if exists ( select name from sysobjects where name = 'sp6tpcc_orderstatus' )
drop procedure sp6tpcc_orderstatus

go

create proc sp6tpcc_orderstatus      @w_id      int,
                                     @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 WITH (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
    where
    customer WITH (repeatableread)
    c_last =

@c_last and
and
    c_w_id = @w_id

    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
    where
    customer WITH (repeatableread)
    c_id = @c_id

    c_d_id = @d_id

    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
    where
    orders WITH (serializable)
    o_c_id = @c_id and
    o_d_id = @d_id and
    o_w_id = @w_id
    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
    where
    order_line WITH (repeatableread)
    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
if exists ( select name from sysobjects where name = 'sp7tpcc_orderstatus' )
    drop procedure sp7tpcc_orderstatus
go

create proc sp7tpcc_orderstatus
    @w_id int,
    @d_id tinyint,

    @c_id int,
    @c_last char(16)

= ""

as

declare @c_balance numeric(12,2),

```

```

and
        c_d_id = @d_id
        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 WITH (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 WITH (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,

```

```

@e_first,
@c_middle,
@o_entry_d,
@o_carrier_id,
@c_balance,
@o_id

go

-----
create deliveryproc.sql
-----
-- File: DELIVERY.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.42
-- Copyright Microsoft, 2003
--
-- - 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 int,
                           @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,
@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 WITH (serializable),

        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
            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
      = @w_id and
      = @d_id and
      = @o_id
      where ol_w_id
            ol_d_id
            ol_o_id

-----
-- accumulate lineitem amounts for this order into customer
-----
update customer
set c_balance =
c_balance + @total,
      c_delivery_cnt
      = c_delivery_cnt + 1
      = @w_id and
      = @d_id and
      = @c_id
      where c_w_id
            c_d_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,

```

```

@oid6,
@oid7,
@oid8,
@oid9,
@oid10

go
if exists (select name from sysobjects where name = 'sp0tpcc_delivery' )
drop procedure sp0tpcc_delivery
go

create proc sp0tpcc_delivery @w_id int,
                             @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 WITH (serializable,

where no_w_id = @w_id and
      no_d_id = @d_id
order by no_o_id asc

updlock)

if (@@rowcount <> 0)

```

```

begin
-----
-- claim the order for this district
-----
delete new_order
where no_w_id = @w_id
      no_d_id = @d_id
      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
      = @w_id and
      = @d_id and
      = @o_id
      where o_w_id
            o_d_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
      = @w_id and
      = @d_id and
      = @o_id
      where ol_w_id
            ol_d_id
            ol_o_id

-----
-- accumulate lineitem amounts for this order into customer
-----
update customer
set c_balance =
c_balance + @total,
      c_delivery_cnt
      = c_delivery_cnt + 1
      = @w_id and
      = @d_id and
      where c_w_id
            c_d_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

if exists (select name from sysobjects where name = 'sp1tpcc_delivery' )
    drop procedure sp1tpcc_delivery
go

create proc sp1tpcc_delivery
    @w_id int,
    @o_carrier_id

smallint
as

declare @d_id tinyint,
    @o_id int,
    @c_id int,
    @total numeric(12,2),

```

```

    @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 WITH (serializable),
            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
            where o_w_id
                = @w_id and

```

```

            o_d_id
            = @d_id and
            = @o_id

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

            update order_line
            set ol_delivery_d =
                @total
            where ol_w_id
                = @w_id and
                ol_d_id
                = @d_id and
                ol_o_id
                = @o_id

            -----
            -- accumulate lineitem amounts for this order into customer
            -----

            update customer
            set c_balance =
                c_balance + @total,
                c_delivery_cnt
            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
if exists (select name from sysobjects where name = 'sp2tpcc_delivery' )
    drop procedure sp2tpcc_delivery
go

create proc sp2tpcc_delivery      @w_id      int,
                                @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

```

```

                                from      new_order WITH (serializable,
                                uplock)
                                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,
                                            = 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 + ol_amount
                                where      ol_w_id   =
                                            @w_id and
                                            ol_d_id   =
                                            @d_id and
                                            ol_o_id   =
                                            @o_id

                                -----
                                -- accumulate lineitem amounts for this order into customer
                                -----

                                update      customer

```

```

                                set          c_balance =
                                c_balance + @total,
                                = 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
if exists (select name from sysobjects where name = 'sp3tpcc_delivery' )
    drop procedure sp3tpcc_delivery
go

create proc sp3tpcc_delivery      @w_id      int,
                                @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 WITH (serializable,
                                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,
        = o_c_id

        where o_w_id

        = @w_id and

        = @d_id and

        = @o_id

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

        update order_line
        set ol_delivery_d =

        getdate(),

        = @total + ol_amount

        where ol_w_id

        = @w_id and

        = @d_id and

        = @o_id

        -----
        -- accumulate lineitem amounts for this order into customer
        -----

        update customer
        set c_balance =

        c_balance + @total,

        = c_delivery_cnt + 1

        where c_w_id

        = @w_id and

        = @d_id and

        = @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

        if exists (select name from sysobjects where name = 'sp4tpcc_delivery' )
                drop procedure sp4tpcc_delivery

        go

        create proc sp4tpcc_delivery @w_id int,
                                     @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 WITH (serializable,
        uplock)
        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

            where
            ol_w_id

            = @total + ol_amount
            ol_w_id

            = @w_id and
            ol_d_id

            = @d_id and

```

```

        ol_o_id

        = @o_id

        -----
        -- accumulate 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
        if exists (select name from sysobjects where name = 'sp5tpcc_delivery')
            drop procedure sp5tpcc_delivery

        go

```

```

create proc sp5tpcc_delivery @w_id int,
        @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 WITH (serializable,
                uplock)
                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,
o_c_id
= o_c_id
= @w_id and
= @d_id and
= @o_id
where o_w_id
o_d_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
ol_d_id
ol_o_id
= @w_id and
= @d_id and
= @o_id
```

```
-----
-- accumulate lineitem amounts for this order into customer
-----
```

```
update customer
set c_balance =
c_balance + @total,
c_delivery_cnt
= c_delivery_cnt + 1
= @w_id and
= @d_id and
= @c_id
where c_w_id
c_d_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,
@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
if exists (select name from sysobjects where name = 'sp6tpcc_delivery')
drop procedure sp6tpcc_delivery
go
create proc sp6tpcc_delivery @w_id int,
@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 WITH (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,
o_c_id
= o_c_id
= @w_id and
= @d_id and
= @o_id
where o_w_id
o_d_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

-----
-- accumulate lineitem amounts for this order into customer
-----
                                update      customer
                                set          c_balance =
c_balance + @total,
                                c_delivery_cnt
                                where        c_w_id
                                = @w_id and
                                = @d_id and
                                = @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
if exists (select name from sysobjects where name = 'sp7tpcc_delivery' )
drop procedure sp7tpcc_delivery
go

create proc sp7tpcc_delivery      @w_id      int,
                                @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 (@@rowcount < 10)
                                begin

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

                                        select      top 1
                                                @o_id      = no_o_id
                                        from          new_order WITH (serializable,

                                        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
-----

```

```

                                where      new_order      = @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

-----
-- accumulate 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

```

createpaymentproc.sql

```

-----
-- File: PAYMENT.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.42
-- Copyright Microsoft, 2003
--
-- - 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 int,
                        @c_w_id int,
                        @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 int,

```

```

        @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 WITH (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 WITH (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,

        where
        @c_id_local = c_id
        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) +
458)                                substring(@data, 1,

-----
-- 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_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

if exists (select name from sysobjects where name = 'sp0tpcc_payment' )
drop procedure sp0tpcc_payment
go

create proc sp0tpcc_payment      @w_id      int,
                                @c_w_id      int,
                                @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   int,
@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 WITH (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 WITH (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_h_amount,

         c_payment_cnt      = c_payment_cnt + 1,
         c_ytd_payment      = c_ytd_payment +
@c_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 +
@c_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 +
@c_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,

```

```

where      @w_id_local = w_id
           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

```

```

if exists (select name from sysobjects where name = 'sp1tpcc_payment' )
drop procedure sp1tpcc_payment

```

```

go

```

```

create proc sp1tpcc_payment      @w_id      int,
                                @c_w_id     int,
                                @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 int,

```

```

                        @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 WITH (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 WITH (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,

        where
        @c_id_local = c_id
        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) +
458)                                substring(@data, 1,

-----
-- 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_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
if exists (select name from sysobjects where name = 'sp2tpcc_payment' )
drop procedure sp2tpcc_payment
go

create proc sp2tpcc_payment      @w_id      int,
                                @c_w_id      int,
                                @h_amount    int,
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 int,
            @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 WITH (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 WITH (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_h_amount,
            c_payment_cnt = c_payment_cnt + 1,
@c_h_amount,
            c_ytd_payment = c_ytd_payment +
@c_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) +
substr(@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 +
@c_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 +
@c_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

```

```

if exists (select name from sysobjects where name = 'sp3tpcc_payment' )
go

create proc sp3tpcc_payment @w_id int,
@c_w_id int,
@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 int,
@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 WITH (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 WITH (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,

        where @c_id_local = c_id
              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_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
if exists (select name from sysobjects where name = 'sp4tpcc_payment' )
        drop procedure sp4tpcc_payment

go

create proc sp4tpcc_payment      @w_id      int,
                                @c_w_id      int,
                                @h_amount      int,

        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 int,
            @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 WITH (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      customer WITH (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_h_amount,
            c_payment_cnt = c_payment_cnt + 1,
@c_h_amount,
            c_ytd_payment = c_ytd_payment +

            @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,

            where     @c_id_local = c_id
                    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) +
substrng(@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 = substrng
(@c_data,1,200)
end

```

```

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

```

```

update      district
set         d_ytd = d_ytd +
@c_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 +
@c_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
if exists (select name from sysobjects where name = 'sp5tpcc_payment' )
drop procedure sp5tpcc_payment

go
create proc sp5tpcc_payment      @w_id      int,
                                @c_w_id      int,
                                @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    int,
        @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 WITH (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 WITH (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 -
@c_h_amount,
            c_payment_cnt = c_payment_cnt + 1,
            c_ytd_payment = c_ytd_payment +
@c_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,

        where
        @c_id_local = c_id
        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) +
458)                                substring(@data, 1,

-----
-- 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_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
if exists (select name from sysobjects where name = 'sp6tpcc_payment' )
        drop procedure sp6tpcc_payment
go

create proc sp6tpcc_payment      @w_id      int,
                                @c_w_id      int,
                                @h_amount    int,
                                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 int,
               @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 WITH (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 WITH (repeatableread)

```

```

where       @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_h_amount,
            c_payment_cnt = c_payment_cnt + 1,
@c_h_amount,
            c_ytd_payment = c_ytd_payment +
            @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) +
substrings(@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 = substrings
(@c_data,1,200)
end

```

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

```

update      district
set         d_ytd      = d_ytd +
@c_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 +
@c_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
if exists (select name from sysobjects where name = 'sp7tpcc_payment' )
    drop procedure sp7tpcc_payment
go

```

```

create proc sp7tpcc_payment      @w_id      int,
                                @c_w_id      int,
                                @h_amount

                                @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   int,
        @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 WITH (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 WITH (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_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

createstocklevproc.sql

```

-----
--
-- File: STOCKLEV.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.42
-- Copyright Microsoft, 2003
--
-- - 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      int,
                                @d_id      tinyint,
                                @threshold 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    < @threshold

go

if exists (select name from sysobjects where name = 'sp0tpcc_stocklevel' )
drop procedure sp0tpcc_stocklevel

go

create proc sp0tpcc_stocklevel    @w_id      int,
                                   @d_id      tinyint,
                                   @threshold 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    < @threshold

go

```

```

if exists (select name from sysobjects where name = 'sp1tpcc_stocklevel' )
drop procedure sp1tpcc_stocklevel

go

create proc sp1tpcc_stocklevel    @w_id      int,
                                   @d_id      tinyint,
                                   @threshold 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    < @threshold

go

if exists (select name from sysobjects where name = 'sp2tpcc_stocklevel' )
drop procedure sp2tpcc_stocklevel

go

create proc sp2tpcc_stocklevel    @w_id      int,
                                   @d_id      tinyint,
                                   @threshold 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

```

```

           s_w_id      = @w_id and
           s_i_id      = ol_i_id and
           s_quantity    < @threshold

go

if exists (select name from sysobjects where name = 'sp3tpcc_stocklevel' )
drop procedure sp3tpcc_stocklevel

go

create proc sp3tpcc_stocklevel    @w_id      int,
                                   @d_id      tinyint,
                                   @threshold 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    < @threshold

go

if exists (select name from sysobjects where name = 'sp4tpcc_stocklevel' )
drop procedure sp4tpcc_stocklevel

go

create proc sp4tpcc_stocklevel    @w_id      int,
                                   @d_id      tinyint,
                                   @threshold 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   < @threshold

go

if exists (select name from sysobjects where name = 'sp5tpcc_stocklevel' )
    drop procedure sp5tpcc_stocklevel
go

create proc sp5tpcc_stocklevel      @w_id      int,
                                   @d_id      tinyint,
                                   @threshold 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   < @threshold

go

if exists (select name from sysobjects where name = 'sp6tpcc_stocklevel' )
    drop procedure sp6tpcc_stocklevel
go

create proc sp6tpcc_stocklevel      @w_id      int,
                                   @d_id      tinyint,
                                   @threshold 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   < @threshold

go

if exists (select name from sysobjects where name = 'sp7tpcc_stocklevel' )
    drop procedure sp7tpcc_stocklevel
go

create proc sp7tpcc_stocklevel      @w_id      int,
                                   @d_id      tinyint,
                                   @threshold 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   < @threshold

go

```

TPCC.H

```

//      File:      TPCC.H
//
//      Ver. 4.41
//
//      Copyright Microsoft,
//      1996, 1997, 1998, 1999, 2000, 2001
//      Purpose:   Header file for TPC-C database loader

// Build number of TPC Benchmark Kit
#define TPCKIT_VER      "4.41"

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

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

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

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

// Default loader arguments
#define BATCH      10000
#define DEFLDPACKSIZE      32768
#define LOADER_RES_FILE      "C:\\MSTPCC.440\\SETUP\\logs\\load.out"

```

```

#define LOG_PATH
"C:\\MSTPCC.440\\SETUP\\LOGS\\";
#define LOADER_NURAND_C
#define DEF_STARTING_WAREHOUSE 1
#define BUILD_INDEX 1
// build both data and indexes
#define INDEX_ORDER 1
// build indexes before load
#define SCALE_DOWN 0
// build a normal scale database
#define INDEX_SCRIPT_PATH "scripts"

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

```

```

long scale_down;
char *index_script_path;
} TPCCCLDR_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();

```

Appendix C Tunable Parameters

Disabled Windows Services

Alerter
Automatic Updates
Computer Browser
Cryptographic Services
DHCP Client
Distributed File System
Distributed Link Tracking Client
DNS Client
Error Reporting Service
HID Input Service
IPSEC Policy Agent
IMAPI CD-Burning COM Service
Kerberos Key Distribution Center
License Logging Service
Messenger
Microsoft Search
Print Spooler
Remote Registry Service
Removable Storage
Secondary Logon
System Event Notification
Task Scheduler
Telnet
Terminal Services Session Directory
WebClient
Windows Audio
Wireless Configuration

Server System Configuratiopn

System Information report written at: 04/22/03 09:21:13
System Name: SQLORCA
[System Summary]

Item	Value
OS Name	Microsoft(R) Windows(R) Server 2003, Datacenter Edition

Version	5.2.3790	Build 3790
OS Manufacturer	Microsoft Corporation	
System Name	SQLORCA	

System Manufacturer	HP
System Model	CELL000
System Type	Itanium (TM) -based System
Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz
-----------	--

Processor ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz

Processor ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz

Processor ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz

Processor ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz

Processor ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz

Processor ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz

Processor ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz

Processor ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz

Processor ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz

Processor ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz

Processor ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz

Processor ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz

Processor ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz

BIOS Version/Date HP 00.101, Not Available

SMBIOS Version 2.3

Windows Directory C:\WINDOWS.2

System Directory C:\WINDOWS.2\system32

Boot Device \Device\HarddiskVolume1

Locale United States

Hardware Abstraction Layer Version = "5.2.3790.2 (dnsrv_dev(v-thief).030401-1637)"

User Name Not Available

Time Zone Pacific Daylight Time

Total Physical Memory 523,261.27 MB

Available Physical Memory 498.99 GB

Total Virtual Memory 1,019.52 GB

Available Virtual Memory 999.37 GB

Page File Space 508.53 GB

Page File C:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

I/O Port 0x0000A000-0x0000AFFF PCI bus

I/O Port 0x0000A000-0x0000AFFF PCI bus

I/O Port 0x0000A000-0x0000AFFF PCI bus

I/O Port 0x0000A000-0x0000AFFF PCI bus

I/O Port 0x0000A000-0x0000AFFF PCI bus

I/O Port 0x0000A000-0x0000AFFF PCI bus

I/O Port 0x0000A000-0x0000AFFF PCI bus

Memory Address 0xF0000000-0xFFFFFFFF PCI bus

Memory Address 0xF0000000-0xFFFFFFFF Emulex LP9802 Fibre

Channel Host Adapter

Memory Address 0xF0000000-0xFFFFFFFF PCI bus

Memory Address 0xF0000000-0xFFFFFFFF QLogic QLA23xx PCI

Fibre Channel Adapter

Memory Address 0xF0000000-0xFFFFFFFF PCI bus

Memory Address 0xF0000000-0xFFFFFFFF Emulex LP9802 Fibre

Channel Host Adapter

Memory Address 0xF0000000-0xFFFFFFFF PCI bus

Memory Address 0xF0000000-0xFFFFFFFF QLogic QLA23xx PCI

Fibre Channel Adapter

Memory Address 0xF0000000-0xFFFFFFFF PCI bus

Memory Address 0xF0000000-0xFFFFFFFF QLogic QLA23xx PCI

Fibre Channel Adapter

Memory Address 0xF0000000-0xFFFFFFFF PCI bus

Memory Address 0xF0000000-0xFFFFFFFF QLogic QLA23xx PCI

Fibre Channel Adapter

Memory Address 0xF0000000-0xFFFFFFFF PCI bus

Memory Address 0xF0000000-0xFFFFFFFF QLogic QLA23xx PCI

Fibre Channel Adapter

Memory Address 0xF0000000-0xFFFFFFFF PCI bus

Memory Address 0xF0000000-0xFFFFFFFF QLogic QLA23xx PCI

Fibre Channel Adapter

Memory Address 0xF0000000-0xFFFFFFFF PCI bus

Memory Address 0xF0000000-0xFFFFFFFF QLogic QLA23xx PCI

Fibre Channel Adapter

Memory Address 0xF0000000-0xFFFFFFFF PCI bus

Memory Address 0xF0000000-0xFFFFFFFF QLogic QLA23xx PCI

I/O Port 0x00000000-0x00000FFF PCI bus

I/O Port 0x00000000-0x00000FFF PCI bus

I/O Port 0x00000000-0x00000FFF PCI bus

I/O Port 0x00000000-0x00000FFF PCI bus

I/O Port 0x00000000-0x00000FFF PCI bus

I/O Port 0x00000000-0x00000FFF PCI bus

I/O Port 0x00000000-0x00000FFF PCI bus

I/O Port 0x00000000-0x00000FFF PCI bus

I/O Port 0x00002000-0x00002FFF PCI bus

I/O Port 0x00002000-0x00002FFF LSI Logic 53C1010-66 Device

I/O Port 0x00002000-0x00002FFF PCI bus

I/O Port 0x00002000-0x00002FFF PCI bus

I/O Port 0x00002000-0x00002FFF PCI bus

I/O Port 0x00002000-0x00002FFF PCI bus

I/O Port 0x00002000-0x00002FFF PCI bus

I/O Port 0x00002000-0x00002FFF PCI bus

I/O Port 0x00002000-0x00002FFF PCI bus

Memory Address 0xD8000000-0xFFFFFFFF PCI bus

Memory Address 0xD8000000-0xFFFFFFFF PCI bus

Memory Address 0xD8000000-0xFFFFFFFF PCI bus

Memory Address 0xD8000000-0xFFFFFFFF PCI bus

Memory Address 0xD8000000-0xFFFFFFFF PCI bus

Memory Address 0xD8000000-0xFFFFFFFF PCI bus

Memory Address 0xD8000000-0xFFFFFFFF PCI bus

Memory Address 0xD8000000-0xFFFFFFFF PCI bus

Memory Address 0xD8000000-0xFFFFFFFF PCI bus

Memory Address 0xD8000000-0xFFFFFFFF PCI bus

Memory Address 0xD8000000-0xFFFFFFFF PCI bus

Memory Address 0xD8000000-0xFFFFFFFF PCI bus

Memory Address 0xD8000000-0xFFFFFFFF PCI bus

Memory Address 0xD8000000-0xFFFFFFFF PCI bus

Memory Address 0xD8000000-0xFFFFFFFF PCI bus

I/O Port 0x00009000-0x00009FFF PCI bus

I/O Port 0x00009000-0x00009FFF PCI bus

I/O Port 0x00009000-0x00009FFF PCI bus

I/O Port 0x00009000-0x00009FFF PCI bus

I/O Port 0x00009000-0x00009FFF PCI bus

I/O Port 0x00009000-0x00009FFF PCI bus

I/O Port 0x00009000-0x00009FFF QLogic QLA23xx PCI Fibre Channel

Adapter

I/O Port 0x00009000-0x00009FFF PCI bus

I/O Port 0x00009000-0x00009FFF PCI bus

I/O Port 0x00006000-0x00007FFF PCI bus

I/O Port 0x00006000-0x00007FFF Emulex LP9802 Fibre Channel Host

Adapter

I/O Port 0x00006000-0x00007FFF PCI bus

I/O Port 0x00006000-0x00007FFF Emulex LP9802 Fibre Channel Host

Adapter

I/O Port 0x00006000-0x00007FFF PCI bus

I/O Port 0x00006000-0x00007FFF PCI bus

I/O Port 0x00006000-0x00007FFF PCI bus

I/O Port 0x00006000-0x00007FFF PCI bus

I/O Port 0x00006000-0x00007FFF PCI bus

[illegible]

0xF0001000-0xF00010FF Emulex LP9802 Fibre Channel Host Adapter
OK
0xF0001000-0xF00010FF Emulex LP9802 Fibre Channel Host Adapter
OK

[Components]

[Multimedia]

[Audio Codecs]

CODEC	Manufacturer	Description	Status	File	Version
		Size	Creation Date		
c:\windows.2\	system32\imaadp32.acm	Microsoft Corporation	OK	C:\WINDOWS.2\system32\IMAADP32.ACM	
	5.2.3783.0 (srv03_rtm.030305-2003)	55.00 KB (56,320 bytes)	3/7/2003 9:10 AM		
c:\windows.2\	system32\msgsm32.acm	Microsoft Corporation	OK	C:\WINDOWS.2\system32\MSGSM32.ACM	
	5.2.3783.0 (srv03_rtm.030305-2003)	66.50 KB (68,096 bytes)	3/7/2003 9:10 AM		
c:\windows.2\	system32\msadp32.acm	Microsoft Corporation	OK	C:\WINDOWS.2\system32\MSADP32.ACM	
	5.2.3783.0 (srv03_rtm.030305-2003)	49.00 KB (50,176 bytes)	3/7/2003 9:10 AM		
c:\windows.2\	system32\tssoft32.acm	DSP GROUP, INC.	OK	C:\WINDOWS.2\system32\TSOFT32.ACM	
	1.01	29.00 KB (29,696 bytes)	3/7/2003 9:11 AM		
c:\windows.2\	system32\msg711.acm	Microsoft Corporation	OK	C:\WINDOWS.2\system32\MSG711.ACM	
	5.2.3783.0 (srv03_rtm.030305-2003)	33.00 KB (33,792 bytes)	3/7/2003 9:10 AM		

[Video Codecs]

CODEC	Manufacturer	Description	Status	File	Version
		Size	Creation Date		
c:\windows.2\	system32\msvidc32.dll	Microsoft Corporation	OK	C:\WINDOWS.2\system32\MSVIDC32.DLL	
	5.2.3783.0 (srv03_rtm.030305-2003)	67.00 KB (68,608 bytes)	3/7/2003 9:10 AM		
c:\windows.2\	system32\mrle32.dll	Microsoft Corporation	OK	C:\WINDOWS.2\system32\MSRLE32.DLL	
	5.2.3783.0 (srv03_rtm.030305-2003)	24.50 KB (25,088 bytes)	3/7/2003 9:10 AM		

[CD-ROM]

Item	Value
Drive	F:
Description	CD-ROM Drive
Media Loaded	No
Media Type	CD-ROM
Name	HP DVD-ROM 305 SCSI CdRom Device
Manufacturer	(Standard CD-ROM drives)
Status	OK
Transfer Rate	Not Available
SCSI Target ID	3
PNP Device ID	SCSI\CDROM&VEN_HP&PROD_DVD-ROM_305&REV_1.01\5&6459022&0&030
Driver	c:\windows.2\system32\drivers\cdrom.sys (5.2.3783.0 (srv03_rtm.030305-2003), 143.50 KB (146,944 bytes), 3/7/2003 9:10 AM)

[Sound Device]

Item	Value
------	-------

[Display]

Item	Value
------	-------

[Infrared]

Item	Value
------	-------

[Input]

[Keyboard]

Item	Value
------	-------

[Pointing Device]

Item	Value
------	-------

[Modem]

Item	Value
------	-------

[Network]

[Adapter]

Item	Value
Name	[00000001] Intel(R) PRO/1000 T Server Adapter
Adapter Type	Ethernet 802.3

Item	Value
Installed	Yes
Adapter Type	Ethernet 802.3
Product Type	Intel(R) PRO/1000 T Server Adapter
PNP Device ID	PCI\VEN_8086&DEV_1004&SUBSYS_10048086&REV_02\4&33B0E0D6&0&08
Last Reset	4/22/2003 9:09 AM
Index	1
Service Name	E1000
IP Address	0.0.0.0
IP Subnet	0.0.0.0
Default IP Gateway	Not Available
DHCP Enabled	Yes
DHCP Server	
DHCP Lease Expires	Not Available
DHCP Lease Obtained	Not Available
MAC Address	00:03:47:72:A1:FF
Memory Address	0xC8000000-0xCFFFFFFF
Memory Address	0xC8020000-0xC802FFFF
IRQ Channel	IRQ 627
Driver	c:\windows.2\system32\drivers\e1000645.sys (6.3.6.31 built by: WinDDK, 363.00 KB (371,712 bytes), 3/7/2003 9:35 AM)

Item	Value
Name	[00000002] Intel(R) PRO/1000 T Server Adapter
Adapter Type	Ethernet 802.3
Product Type	Intel(R) PRO/1000 T Server Adapter
Installed	Yes
PNP Device ID	PCI\VEN_8086&DEV_1004&SUBSYS_10048086&REV_02\4&17A9C6C8&0&08
Last Reset	4/22/2003 9:09 AM
Index	2
Service Name	E1000
IP Address	15.1.100.11
IP Subnet	255.255.255.0
Default IP Gateway	Not Available
DHCP Enabled	Yes
DHCP Server	15.1.100.1
DHCP Lease Expires	4/30/2003 9:14 AM
DHCP Lease Obtained	4/22/2003 9:14 AM
MAC Address	00:03:47:72:A1:BE
Memory Address	0xC8000000-0xCFFFFFFF
Memory Address	0xC8020000-0xC802FFFF
IRQ Channel	IRQ 35
Driver	c:\windows.2\system32\drivers\e1000645.sys (6.3.6.31 built by: WinDDK, 363.00 KB (371,712 bytes), 3/7/2003 9:35 AM)

Item	Value
Name	[00000003] Broadcom NetXtreme Gigabit Ethernet
Adapter Type	Ethernet 802.3
Product Type	Broadcom NetXtreme Gigabit Ethernet
Installed	Yes
PNP Device ID	PCI\VEN_14E4&DEV_1645&SUBSYS_128A103C&REV_15\4&137A5C4D&0&08
Last Reset	4/22/2003 9:09 AM

Index 3
Service Name b57nd
IP Address 10.193.24.141
IP Subnet 255.255.255.0
Default IP Gateway 10.193.24.1
DHCP Enabled Yes
DHCP Server 10.193.12.10
DHCP Lease Expires 4/30/2003 9:14 AM
DHCP Lease Obtained 4/22/2003 9:14 AM
MAC Address 00:04:76:DF:CE:E3
Memory Address 0xC0000000-0xC7FFFFFFF
IRQ Channel IRQ 90
Driver c:\windows.2\system32\drivers\b57xp64.sys (2.91.0.0 built by: WinDDK, 499.25 KB (511,232 bytes), 3/7/2003 9:36 AM)

Name [00000004] RAS Async Adapter
Adapter TypeNot Available
Product TypeRAS Async Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 4/22/2003 9:09 AM
Index 4
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000005] WAN Miniport (L2TP)
Adapter TypeNot Available
Product TypeWAN Miniport (L2TP)
Installed Yes
PNP Device ID ROOT\MS_L2TPMINI\PORT\0000
Last Reset 4/22/2003 9:09 AM
Index 5
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000006] WAN Miniport (PPTP)
Adapter TypeWide Area Network (WAN)
Product TypeWAN Miniport (PPTP)

PNP Device IDYes ROOT\MS_PPTP\MINI\PORT\0000
Last Reset 4/22/2003 9:09 AM
Index 6
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30

Name [00000007] WAN Miniport (PPPOE)
Adapter TypeWide Area Network (WAN)
Product TypeWAN Miniport (PPPOE)
Installed Yes
PNP Device ID ROOT\MS_PPPOE\MINI\PORT\0000
Last Reset 4/22/2003 9:09 AM
Index 7
Service Name RasPppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30

Name [00000008] Direct Parallel
Adapter TypeNot Available
Product TypeDirect Parallel
Installed Yes
PNP Device ID ROOT\MS_PT\MINI\PORT\0000
Last Reset 4/22/2003 9:09 AM
Index 8
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000009] WAN Miniport (IP)
Adapter TypeNot Available
Product TypeWAN Miniport (IP)
Installed Yes
PNP Device ID ROOT\MS_NDIS\WANIP\0000
Last Reset 4/22/2003 9:09 AM

Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

[Protocol]

Item	Value
Name	MSAFD Tcpip [TCP/IP]
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	16 bytes
Maximum Message Size	0 bytes
Message Oriented	No
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	Yes
Supports Graceful Closing	Yes
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Item	Value
Name	MSAFD Tcpip [UDP/IP]
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	16 bytes
Maximum Message Size	63.93 KB (65,467 bytes)
Message Oriented	Yes
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	Yes
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	Yes

Item	Value
Name	RSVP UDP Service Provider
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No

Maximum Address Size 16 bytes
Maximum Message Size 63.93 KB (65,467 bytes)
Message Oriented Yes
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption Yes
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting Yes

Name RSVP TCP Service Provider
Connectionless Service No
Guarantees Delivery Yes
Guarantees Sequencing Yes
Maximum Address Size 16 bytes
Maximum Message Size 0 bytes
Message Oriented No
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption Yes
Supports Expedited Data Yes
Supports Graceful Closing Yes
Supports Guaranteed Bandwidth No
Supports Multicasting No

[WinSock]

Item Value
File c:\windows.2\system32\wsock32.dll
Size 23.00 KB (23,552 bytes)
Version 5.2.3783.0 (srv03_rtm.030305-2003)

[Ports]

[Serial]

Item Value

[Parallel]

Item Value

[Storage]

[Drives]

Item Value
Drive C:
Description Local Fixed Disk
Compressed No
File System FAT32
Size 16.59 GB (17,815,453,696 bytes)
Free Space 9.57 GB (10,272,800,768 bytes)
Volume Name
Volume Serial Number 4414B961

Drive E:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 16.85 GB (18,095,611,904 bytes)
Free Space 3.50 GB (3,756,204,032 bytes)
Volume Name New Volume
Volume Serial Number 20DD9082

Drive F:
Description CD-ROM Disc

Drive G:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 16.83 GB (18,075,348,992 bytes)
Free Space 6.75 GB (7,248,465,920 bytes)
Volume Name TPC-C
Volume Serial Number 28E4DDC7

Drive L:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available

Drive Z:
Description Network Connection
Provider Name \\ssd\\lab\\d\$

[Disks]

Item Value
Description Disk drive
Manufacturer (Standard disk drives)

Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 25
SCSI Target ID 0
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #64, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #64, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #64, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 25
SCSI Target ID 1
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #65, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #65, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #65, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)

Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 25
 SCSI Target ID 2
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #66, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #66, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #66, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 25
 SCSI Target ID 3
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #67, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #67, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #67, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 3
 SCSI Target ID 0
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #4, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #4, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #4, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 3
 SCSI Target ID 1
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #5, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #5, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #5, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 3
 SCSI Target ID 2
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #6, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #6, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #6, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 3
 SCSI Target ID 3
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #7, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #7, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #7, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 6
 SCSI Target ID 0
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #16, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #16, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #16, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 6
 SCSI Target ID 1
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #17, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #17, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #17, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)

Partition Starting Offset 92,673,146,880 bytes
 Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 6
 SCSI Target ID 2
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #18, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #18, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #18, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 6
 SCSI Target ID 3
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #19, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #19, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #19, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)

Partition Starting Offset 92,673,146,880 bytes
 Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 4
 SCSI Target ID 0
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #8, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #8, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #8, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 4
 SCSI Target ID 1
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #9, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #9, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #9, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)

Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 4
SCSI Target ID 2
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #10, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #10, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #10, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 18
SCSI Target ID 0
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #48, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #48, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes

Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 18
SCSI Target ID 1
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #49, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #49, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #49, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 18
SCSI Target ID 2
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #50, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #50, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes

Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 18
SCSI Target ID 3
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #51, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #51, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #51, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 5
SCSI Target ID 0
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #12, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #12, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes

Partition Disk #12, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 5
SCSI Target ID 1
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #13, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #13, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #13, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 5
SCSI Target ID 2
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #14, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #14, Partition #1

Partition Starting Offset (33,554,432,000 bytes)
Partition Disk #14, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 5
SCSI Target ID 3
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #15, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #15, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #15, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 17
SCSI Target ID 0
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #44, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #44, Partition #1

Partition Starting Offset (33,554,432,000 bytes)
Partition Disk #44, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 17
SCSI Target ID 1
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #45, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #45, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #45, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 17
SCSI Target ID 2
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #46, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #46, Partition #1

Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #46, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 17
SCSI Target ID 3
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #47, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #47, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #47, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 15
SCSI Target ID 0
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #36, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)

Partition Starting Offset 92,673,146,880 bytes
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #36, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 15
SCSI Target ID 1
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #37, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #37, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #37, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 15
SCSI Target ID 2
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #38, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)

Partition Starting Offset 92,673,146,880 bytes
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #38, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 15
SCSI Target ID 3
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #39, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #39, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #39, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 9
SCSI Target ID 0
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #20, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)

Partition Starting Offset 136,314,880 bytes
Partition Disk #20, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #20, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 9
SCSI Target ID 1
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #21, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #21, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #21, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 9
SCSI Target ID 2
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255

Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #22, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #22, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 9
SCSI Target ID 3
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #23, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #23, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #23, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 12
SCSI Target ID 0
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255

Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #32, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #32, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 12
SCSI Target ID 1
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255
Partition Disk #33, Partition #0
Partition Size 54.93 GB (58,982,400,000 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #33, Partition #1
Partition Size 31.25 GB (33,554,432,000 bytes)
Partition Starting Offset 59,118,714,880 bytes
Partition Disk #33, Partition #2
Partition Size 135.74 GB (145,752,064,000 bytes)
Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 12
SCSI Target ID 2
Sectors/Track 63
Size 237.41 GB (254,917,877,760 bytes)
Total Cylinders 30,992
Total Sectors 497,886,480
Total Tracks 7,902,960
Tracks/Cylinder 255

Partition Disk #34, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #34, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #34, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 12
 SCSI Target ID 3
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #35, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #35, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #35, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 16
 SCSI Target ID 0
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480

Tracks/Cylinder 255
 Partition Disk #40, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #40, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #40, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 16
 SCSI Target ID 1
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #41, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #41, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #41, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 16
 SCSI Target ID 2
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480

Tracks/Cylinder 255
 Partition Disk #42, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #42, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #42, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 16
 SCSI Target ID 3
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #43, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #43, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #43, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 23
 SCSI Target ID 0
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480

Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #56, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #56, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #56, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 23
 SCSI Target ID 1
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #57, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #57, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #57, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 23
 SCSI Target ID 2
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)

Total Sectors 497,886,480 30,992
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #58, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #58, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #58, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 23
 SCSI Target ID 3
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #59, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #59, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #59, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 22
 SCSI Target ID 0
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)

Total Sectors 497,886,480 30,992
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #52, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #52, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #52, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 22
 SCSI Target ID 1
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #53, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #53, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #53, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 22
 SCSI Target ID 2
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)

Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #54, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #54, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #54, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 22
 SCSI Target ID 3
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #55, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #55, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #55, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 10
 SCSI Target ID 0

Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #24, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #24, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #24, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 10
 SCSI Target ID 1
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #25, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #25, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #25, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 10
 SCSI Target ID 2

Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #26, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #26, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #26, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 10
 SCSI Target ID 3
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #27, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #27, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #27, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 24
 SCSI Target ID 0

Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #60, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #60, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #60, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 24
 SCSI Target ID 1
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #61, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #61, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #61, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1

~~SCSI Target ID~~ 2
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #62, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #62, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #62, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 24
 SCSI Target ID 3
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #63, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #63, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #63, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1

~~SCSI Target ID~~ 0
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #28, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #28, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #28, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 11
 SCSI Target ID 1
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #29, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #29, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #29, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1

SCSI Port 11
 SCSI Target ID 2
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #30, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #30, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #30, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 11
 SCSI Target ID 3
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #31, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #31, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #31, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available

SCSI Logical Unit 1
 SCSI Port 4
 SCSI Target ID 3
 Sectors/Track 63
 Size 237.41 GB (254,917,877,760 bytes)
 Total Cylinders 30,992
 Total Sectors 497,886,480
 Total Tracks 7,902,960
 Tracks/Cylinder 255
 Partition Disk #11, Partition #0
 Partition Size 54.93 GB (58,982,400,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #11, Partition #1
 Partition Size 31.25 GB (33,554,432,000 bytes)
 Partition Starting Offset 59,118,714,880 bytes
 Partition Disk #11, Partition #2
 Partition Size 135.74 GB (145,752,064,000 bytes)
 Partition Starting Offset 92,673,146,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP A6218A SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 1
 SCSI Logical Unit 1
 SCSI Port 1
 SCSI Target ID 108
 Sectors/Track 63
 Size 499.99 GB (536,864,025,600 bytes)
 Total Cylinders 65,270
 Total Sectors 1,048,562,550
 Total Tracks 16,643,850
 Tracks/Cylinder 255
 Partition Disk #2, Partition #0
 Partition Size 1.00 MB (1,048,576 bytes)
 Partition Starting Offset 17,408 bytes
 Partition Disk #2, Partition #1
 Partition Size 499.87 GB (536,736,659,968 bytes)
 Partition Starting Offset 134,235,136 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP A6218A SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 1
 SCSI Logical Unit 3
 SCSI Port 1

Sectors/Track 63
 Size 433.00 GB (464,925,726,720 bytes)
 Total Cylinders 56,524
 Total Sectors 908,058,060
 Total Tracks 14,413,620
 Tracks/Cylinder 255
 Partition Disk #3, Partition #0
 Partition Size 1.00 MB (1,048,576 bytes)
 Partition Starting Offset 17,408 bytes
 Partition Disk #3, Partition #1
 Partition Size 432.87 GB (464,795,957,760 bytes)
 Partition Starting Offset 134,235,136 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP A6218A SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 1
 SCSI Logical Unit 0
 SCSI Port 0
 SCSI Target ID 110
 Sectors/Track 63
 Size 499.99 GB (536,864,025,600 bytes)
 Total Cylinders 65,270
 Total Sectors 1,048,562,550
 Total Tracks 16,643,850
 Tracks/Cylinder 255
 Partition Disk #0, Partition #0
 Partition Size 1.00 MB (1,048,576 bytes)
 Partition Starting Offset 17,408 bytes
 Partition Disk #0, Partition #1
 Partition Size 499.87 GB (536,736,659,968 bytes)
 Partition Starting Offset 134,235,136 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP A6218A SCSI Disk Device
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus 1
 SCSI Logical Unit 2
 SCSI Port 0
 SCSI Target ID 110
 Sectors/Track 63
 Size 433.00 GB (464,925,726,720 bytes)
 Total Cylinders 56,524
 Total Sectors 908,058,060
 Total Tracks 14,413,620

Tracks/Cylinder 255
Partition Disk #1, Partition #0
Partition Size 1.00 MB (1,048,576 bytes)
Partition Starting Offset 17,408 bytes
Partition Disk #1, Partition #1
Partition Size 432.87 GB (464,795,957,760 bytes)
Partition Starting Offset 134,235,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model HP 18.2G ST318452LC SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 27
SCSI Target ID 2
Sectors/Track 63
Size 16.95 GB (18,202,544,640 bytes)
Total Cylinders 2,213
Total Sectors 35,551,845
Total Tracks 564,315
Tracks/Cylinder 255
Partition Disk #69, Partition #0
Partition Size 1.00 MB (1,048,576 bytes)
Partition Starting Offset 17,408 bytes
Partition Disk #69, Partition #1
Partition Size 16.83 GB (18,075,353,088 bytes)
Partition Starting Offset 134,235,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model HP 36.4G ST336706LC SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 27
SCSI Target ID 0
Sectors/Track 63
Size 33.91 GB (36,413,314,560 bytes)
Total Cylinders 4,427
Total Sectors 71,119,755
Total Tracks 1,128,885
Tracks/Cylinder 255
Partition Disk #68, Partition #0
Partition Size 345.12 MB (361,880,064 bytes)
Partition Starting Offset 32,256 bytes

Partition Size 16.83 GB (18,075,353,088 bytes)
Partition Starting Offset 493,516,800 bytes
Partition Disk #68, Partition #2
Partition Size 16.85 GB (18,095,616,000 bytes)
Partition Starting Offset 18,317,698,560 bytes

[SCSI]

Item Value
Name LSI Logic 53C1010-66 Device
Manufacturer LSI Logic Inc.
Status OK
PNP Device ID PCI\VEN_1000&DEV_0021&SUBSYS_1340103C&REV_01\4&2A694D11&0&08
I/O Port 0x00002000-0x00002FFF
Memory Address 0xD0044000-0xD00443FF
Memory Address 0xD0040000-0xD0041FFF
IRQ Channel IRQ 46
Driver c:\windows.2\system32\drivers\sym_u3.sys (5.2.3673.0
(_.020828-1450), 66.00 KB (67,584 bytes), 3/7/2003 9:11 AM)

Name LSI Logic 53C1010-66 Device
Manufacturer LSI Logic Inc.
Status OK
PNP Device ID PCI\VEN_1000&DEV_0021&SUBSYS_1340103C&REV_01\4&2A694D11&0&09
I/O Port 0x00002100-0x000021FF
Memory Address 0xD0045000-0xD00453FF
Memory Address 0xD0042000-0xD0043FFF
IRQ Channel IRQ 47
Driver c:\windows.2\system32\drivers\sym_u3.sys (5.2.3673.0
(_.020828-1450), 66.00 KB (67,584 bytes), 3/7/2003 9:11 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&38E5DBD97&0&08
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE0000000-0xEFFFFFFF
IRQ Channel IRQ 68
Driver c:\windows.2\system32\drivers\ql2300.sys (8.2.2.10 hp-2 (W64
VI), 690.88 KB (707,456 bytes), 4/17/2003 8:47 AM)

Name Emulex LP9802 Fibre Channel Host Adapter
Manufacturer Emulex
Status OK
PNP Device ID PCI\VEN_10DF&DEV_F980&SUBSYS_F98010DF&REV_01\4&26340CF2&0&08

Memory Address 0xF0000000-0xF00010FF
I/O Port 0x00006000-0x00007FFF
IRQ Channel IRQ 79
Driver c:\windows.2\system32\drivers\lp6nds35.sys (6-4.82a8
09/06/2002 Win .Net 64 bits (srv03_rtm.030305-2003), 225.50 KB (230,912
bytes), 3/7/2003 9:10 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&200C63EC&0&08
I/O Port 0x0000C000-0x0000DFFF
Memory Address 0xE0000000-0xEFFFFFFF
IRQ Channel IRQ 134
Driver c:\windows.2\system32\drivers\ql2300.sys (8.2.2.10 hp-2 (W64
VI), 690.88 KB (707,456 bytes), 4/17/2003 8:47 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&3633CCA9&0&08
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xF0000000-0xFFFFFFFF
IRQ Channel IRQ 145
Driver c:\windows.2\system32\drivers\ql2300.sys (8.2.2.10 hp-2 (W64
VI), 690.88 KB (707,456 bytes), 4/17/2003 8:47 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&3815FDFD&0&08
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE0000000-0xEFFFFFFF
IRQ Channel IRQ 216
Driver c:\windows.2\system32\drivers\ql2300.sys (8.2.2.10 hp-2 (W64
VI), 690.88 KB (707,456 bytes), 4/17/2003 8:47 AM)

Name Emulex LP9802 Fibre Channel Host Adapter
Manufacturer Emulex
Status OK
PNP Device ID PCI\VEN_10DF&DEV_F980&SUBSYS_F98010DF&REV_01\4&255C4D58&0&08
Memory Address 0xF0000000-0xFFFFFFFF
Memory Address 0xF0001000-0xF00010FF
I/O Port 0x00006000-0x00007FFF
IRQ Channel IRQ 227

Driver c:\windows.2\system32\drivers\lp6nds35.sys (6-4.82a8 09/06/2002 Win .Net 64 bits (srv03_rtm.030305-2003), 225.50 KB (230,912 bytes), 3/7/2003 9:10 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&1F34A452&0&08
I/O Port 0x0000C000-0x0000DFFF
Memory Address 0xE0000000-0xEFFFFFFF
IRQ Channel IRQ 282
Driver c:\windows.2\system32\drivers\ql2300.sys (8.2.2.10 hp-2 (W64 VI), 690.88 KB (707,456 bytes), 4/17/2003 8:47 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&355C0D0F&0&08
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xF0000000-0xFFFFFFFF
IRQ Channel IRQ 293
Driver c:\windows.2\system32\drivers\ql2300.sys (8.2.2.10 hp-2 (W64 VI), 690.88 KB (707,456 bytes), 4/17/2003 8:47 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&A03B852&0&08
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE0000000-0xEFFFFFFF
IRQ Channel IRQ 364
Driver c:\windows.2\system32\drivers\ql2300.sys (8.2.2.10 hp-2 (W64 VI), 690.88 KB (707,456 bytes), 4/17/2003 8:47 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&EDA159&0&08
I/O Port 0x0000C000-0x0000DFFF
Memory Address 0xE0000000-0xEFFFFFFF
IRQ Channel IRQ 430
Driver c:\windows.2\system32\drivers\ql2300.sys (8.2.2.10 hp-2 (W64 VI), 690.88 KB (707,456 bytes), 4/17/2003 8:47 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&345102A3&0&08
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xF0000000-0xFFFFFFFF
IRQ Channel IRQ 441
Driver c:\windows.2\system32\drivers\ql2300.sys (8.2.2.10 hp-2 (W64 VI), 690.88 KB (707,456 bytes), 4/17/2003 8:47 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&240E8D59&0&08
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE0000000-0xEFFFFFFF
IRQ Channel IRQ 512
Driver c:\windows.2\system32\drivers\ql2300.sys (8.2.2.10 hp-2 (W64 VI), 690.88 KB (707,456 bytes), 4/17/2003 8:47 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&1551CFD&0&08
I/O Port 0x0000C000-0x0000DFFF
Memory Address 0xF0000000-0xEFFFFFFF
IRQ Channel IRQ 578
Driver c:\windows.2\system32\drivers\ql2300.sys (8.2.2.10 hp-2 (W64 VI), 690.88 KB (707,456 bytes), 4/17/2003 8:47 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&266723A4&0&08
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xF0000000-0xFFFFFFFF
IRQ Channel IRQ 589
Driver c:\windows.2\system32\drivers\ql2300.sys (8.2.2.10 hp-2 (W64 VI), 690.88 KB (707,456 bytes), 4/17/2003 8:47 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK

PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&36A998EE&0&08
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE0000000-0xEFFFFFFF
IRQ Channel IRQ 660
Driver c:\windows.2\system32\drivers\ql2300.sys (8.2.2.10 hp-2 (W64 VI), 690.88 KB (707,456 bytes), 4/17/2003 8:47 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&1DC83F43&0&08
I/O Port 0x0000C000-0x0000DFFF
Memory Address 0xE0000000-0xEFFFFFFF
IRQ Channel IRQ 726
Driver c:\windows.2\system32\drivers\ql2300.sys (8.2.2.10 hp-2 (W64 VI), 690.88 KB (707,456 bytes), 4/17/2003 8:47 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&7AB2207&0&08
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xF0000000-0xEFFFFFFF
IRQ Channel IRQ 737
Driver c:\windows.2\system32\drivers\ql2300.sys (8.2.2.10 hp-2 (W64 VI), 690.88 KB (707,456 bytes), 4/17/2003 8:47 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&24E12DB3&0&08
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE0000000-0xEFFFFFFF
IRQ Channel IRQ 808

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&2173759B&0&08
I/O Port 0x00009000-0x00009FFF
Memory Address 0xC8000000-0xCFFFFFFF
IRQ Channel IRQ 841

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&227BD57&0&08
I/O Port 0x0000C000-0x0000DFFF
Memory Address 0xE0000000-0xEFFFFFFF
IRQ Channel IRQ 874

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&279B1EA1&0&08
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xF0000000-0xFFFFFFFF
IRQ Channel IRQ 885

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&35D6F894&0&08
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE0000000-0xEFFFFFFF
IRQ Channel IRQ 956

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&1CF59EE9&0&08
I/O Port 0x0000C000-0x0000DFFF
Memory Address 0xE0000000-0xEFFFFFFF
IRQ Channel IRQ 1022

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&331D07A6&0&08
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xF0000000-0xFFFFFFFF
IRQ Channel IRQ 1033

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic

Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&7C4B2E9&0&08
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE0000000-0xEFFFFFFF
IRQ Channel IRQ 1104

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&111CA6C2&0&08
I/O Port 0x0000C000-0x0000DFFF
Memory Address 0xE0000000-0xEFFFFFFF
IRQ Channel IRQ 1170

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&3690080C&0&08
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xF0000000-0xFFFFFFFF
IRQ Channel IRQ 1181

[IDE]

Item	Value
------	-------

[Printing]

Name	Driver	Port Name	Server Name
------	--------	-----------	-------------

[Problem Devices]

Device	PNP Device ID	Error Code
PCI Serial Port	PCI\VEN_103C&DEV_1048&SUBSYS_1223103C&REV_02\4&4F5EBC7&0&00	The drivers for this device are not installed.
Ethernet Controller	PCI\VEN_1011&DEV_0019&SUBSYS_104F103C&REV_41\4&4F5EBC7&0&08	The drivers for this device are not installed.

[USB]

Device	PNP Device ID
--------	---------------

[Software Environment]

[System Drivers]

Name	Description	File	Type	Started	Start		
Mode	State	Status	Error Control	Accept	Pause		
Stop	Accept	Pause	Accept				
4mmdat	4mmdat	c:\windows.2\system32\drivers\4mmdat.sys	Kernel Driver	Yes	Manual	Running	OK
abiosdsk	Abiosdsk	Not Available	Kernel Driver	No			Disabled
acpi	Microsoft ACPI Driver	c:\windows.2\system32\drivers\acpi.sys	Kernel Driver	Yes	Manual	Running	OK
acpiec	ACPIEC	c:\windows.2\system32\drivers\acpiec.sys	Kernel Driver	No	Disabled	Stopped	OK
adpu160m	adpu160m	Not Available	Kernel Driver	No			Disabled
adpu320	adpu320	Not Available	Kernel Driver	No			Disabled
afcnt	afcnt	Not Available	Kernel Driver	No			Disabled
afd	AFD Networking Support Environment	c:\windows.2\system32\drivers\afd.sys	Kernel Driver	Yes	Auto	Running	OK
aic78u2	aic78u2	Not Available	Kernel Driver	No			Disabled
aic78xx	aic78xx	Not Available	Kernel Driver	No			Disabled
aliide	AliIde	Not Available	Kernel Driver	No			Disabled
asynccmac	RAS Asynchronous Media Driver	c:\windows.2\system32\drivers\asynccmac.sys	Kernel Driver	No	Manual	Stopped	OK
atapi	atapi	c:\windows.2\system32\drivers\atapi.sys	Kernel Driver	No	Disabled	Stopped	OK
atdisk	Atdisk	Not Available	Kernel Driver	No			Disabled

atmarpc	ATM ARP Client Protocol c:\windows.2\system32\drivers\atmarpc.sys					Kernel	disk	Disk Driver Yes Boot Running OK Normal					ipfilterdriver	IP Traffic Filter Driver c:\windows.2\system32\drivers\ipfltrdrv.sys					Kernel
Driver	No	Manual	Stopped	OK	Normal		dmboot	No Kernel Driver Yes Disabled Running OK Normal					Driver	No	Manual	Stopped	OK	Normal	
audstub	Audio Stub Driver c:\windows.2\system32\drivers\audstub.sys					Kernel	dmio	Logical Disk Manager Driver c:\windows.2\system32\drivers\dmio.sys					Driver	No	No	Stopped	OK	Kernel	
Driver	Yes	Manual	Running	OK	Normal		Driver	Yes	Boot	Running	OK	Normal	ipinip	IP In IP Tunnel Driver c:\windows.2\system32\drivers\ipinip.sys					Kernel
b57nd	Broadcom NetXtreme Gigabit Ethernet c:\windows.2\system32\drivers\b57xp64.sys					Kernel	dmload	dmload c:\windows.2\system32\drivers\dmload.sys Kernel Driver Yes Boot Running OK Normal					Driver	No	Manual	Stopped	OK	Normal	
Driver	Yes	Manual	Running	OK	Normal		e1000	Intel(R) PRO/1000 Device Driver c:\windows.2\system32\drivers\e1000645.sys					Driver	No	No	Stopped	OK	Kernel	
beep	Beep c:\windows.2\system32\drivers\beep.sys Kernel Driver Yes System Running OK Normal						Driver	Yes	Manual	Running	OK	Normal	ipsec	IPSEC driver c:\windows.2\system32\drivers\ipsec.sys Kernel Driver Yes System Running OK Normal					
cbidf2k	cbidf2k c:\windows.2\system32\drivers\cbidf2k.sys Kernel Driver No Disabled Stopped OK Normal						fastfat	Fastfat c:\windows.2\system32\drivers\fastfat.sys File System Driver Yes Disabled Running OK					Driver	Yes	System	Running	OK	Normal	
cdfs	Cdfs c:\windows.2\system32\drivers\cdfs.sys File System Driver Yes Disabled Running OK Normal						fdc	Fdc c:\windows.2\system32\drivers\fdc.sys No System Stopped OK Ignore					Driver	Yes	System	Running	OK	Kernel	
cdrom	CD-ROM Driver c:\windows.2\system32\drivers\cdrom.sys					Kernel	fips	Fips c:\windows.2\system32\drivers\fips.sys Yes System Running OK Normal					Driver	No	Yes	Boot	Running	OK	
Driver	Yes	System	Running	OK	Normal		flypydisk	Flypydisk c:\windows.2\system32\drivers\flypydisk.sys Kernel Driver No System Stopped OK Ignore					Driver	No	Yes	Boot	Running	OK	
changer	Changer Not Available Kernel Driver No Stopped OK Ignore No					System	ftdisk	Volume Manager Driver c:\windows.2\system32\drivers\ftdisk.sys					Driver	Yes	Boot	Running	OK	Kernel	
clusdisk	Cluster Disk Driver c:\windows.2\system32\drivers\clusdisk.sys					Kernel	gpc	Generic Packet Classifier c:\windows.2\system32\drivers\msgpc.sys					Driver	Yes	Manual	Running	OK	Kernel	
Driver	No	Disabled	Stopped	OK	Normal		hpn	hpn Not Available Kernel Driver No Stopped OK Normal No					Driver	No	Yes	Not Available	Kernel Driver	No	
cmdide	CmdIde Not Available Kernel Driver No Stopped OK Normal No					Disabled	http	HTTP c:\windows.2\system32\drivers\http.sys No Manual Stopped OK Normal					Driver	No	No	Not Available	Kernel Driver	No	
cpqarry2	cpqarry2 Not Available Kernel Driver No Stopped OK Normal No					Disabled	i2omgmt	i2omgmt Not Available Kernel Driver No Stopped OK Normal No					System	Driver	Yes	Boot	Running	OK	
cpqcissm	cpqcissm Not Available Kernel Driver No Stopped OK Normal No					Disabled	imapi	CD-Burning Filter Driver c:\windows.2\system32\drivers\imapi.sys					Driver	No	No	System	Stopped	OK	
cpqfcalm	cpqfcalm Not Available Kernel Driver No Stopped OK Normal No					Disabled	intelide	IntelIde Not Available Kernel Driver No Stopped OK Normal No					Disabled	Driver	No	No	File System Driver	Yes	
crcdisk	CRC Disk Filter Driver c:\windows.2\system32\drivers\crcdisk.sys					Kernel							mrxdav	WebDav Client Redirector c:\windows.2\system32\drivers\mrxdav.sys					File
Driver	Yes	Boot	Running	OK	Normal							System Driver	No	Manual	Stopped	OK	Kernel		
dfsdriver	DfsDriver c:\windows.2\system32\drivers\dfs.sys Yes Boot Running OK Normal					File						Driver	Yes	System	Running	OK	Normal		
System Driver	Normal	No	Yes									mrxsmb	MRXSMB c:\windows.2\system32\drivers\mrxsmb.sys File System Driver Yes System Running OK Normal					Running	

msfs	Msfs	c:\windows.2\system32\drivers\msfs.sys	File		
System Driver	Yes	System	Running	OK	
	Normal	No	Yes		
mup	Mup	c:\windows.2\system32\drivers\mup.sys	File		
System Driver	Yes	Boot	Running	OK	
	Normal	No	Yes		
ndis	NDIS System Driver	c:\windows.2\system32\drivers\ndis.sys	Kernel Driver	Yes	
	Boot	Running	OK	Normal	No
ndistapi	Remote Access NDIS TAPI Driver	c:\windows.2\system32\drivers\ndistapi.sys			Kernel
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
ndisuio	NDIS Usermode I/O Protocol	c:\windows.2\system32\drivers\ndisuio.sys			Kernel
Driver	No	Manual	Stopped	OK	Normal
	No	No			
ndiswan	Remote Access NDIS WAN Driver	c:\windows.2\system32\drivers\ndiswan.sys			Kernel
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
ndproxy	NDIS Proxy	c:\windows.2\system32\drivers\ndproxy.sys			
	Kernel Driver	Yes	Manual	Running	OK
	Normal	No	Yes		
netbios	NetBIOS Interface	c:\windows.2\system32\drivers\netbios.sys			File
System Driver	Yes	System	Running	OK	
	Normal	No	Yes		
netbt	NetBios over Tcpip	c:\windows.2\system32\drivers\netbt.sys			Kernel
Driver	Yes	System	Running	OK	Normal
	No	Yes			
nfrd960	nfrd960	Not Available	Kernel Driver	No	Disabled
	Stopped	OK	Normal	No	No
npfs	Npfs	c:\windows.2\system32\drivers\npfs.sys	File		
System Driver	Yes	System	Running	OK	
	Normal	No	Yes		
ntfs	Ntfs	c:\windows.2\system32\drivers\ntfs.sys	File		
System Driver	Yes	Disabled	Running	OK	
	Normal	No	Yes		
null	Null	c:\windows.2\system32\drivers\null.sys	Kernel		
Driver	Yes	System	Running	OK	Normal
	No	Yes			
partmgr	Partition Manager	c:\windows.2\system32\drivers\partmgr.sys			Kernel
Driver	Yes	Boot	Running	OK	Normal
	No	Yes			
pci	PCI Bus Driver	c:\windows.2\system32\drivers\pci.sys	Kernel Driver	Yes	
	Boot	Running	OK	Critical	No
pciide	PCI IDE	Not Available	Kernel Driver	No	Disabled
	Stopped	OK	Normal	No	No
pcmcia	Pcmcia	c:\windows.2\system32\drivers\pcmcia.sys			
	Kernel Driver	No	Disabled	Stopped	OK
	Normal	No	No		
pdcomp	PDCOMP	Not Available	Kernel Driver	No	Manual
	Stopped	OK	Ignore	No	No
pdframe	PDFRAME	Not Available	Kernel Driver	No	Manual
	Stopped	OK	Ignore	No	No
pdreli	PDRELI	Not Available	Kernel Driver	No	Manual
	Stopped	OK	Ignore	No	No
pdrframe	PDRFRAME	Not Available	Kernel Driver	No	Manual
	Stopped	OK	Ignore	No	No
pptpminiport	WAN Miniport (PPTP)	c:\windows.2\system32\drivers\rasppptp.sys			Kernel
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
processor	Processor Driver	c:\windows.2\system32\drivers\processr.sys			Kernel
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
procinfo	ProcInfo	c:\windows.2\system32\drivers\procinfo.sys			
	Kernel Driver	No	Manual	Stopped	OK
	Normal	No	No		
ptilink	Direct Parallel Link Driver	c:\windows.2\system32\drivers\ptilink.sys			Kernel
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
ql1080	ql1080	Not Available	Kernel Driver	No	Disabled
	Stopped	OK	Normal	No	No
ql10wnt	Ql10wnt	Not Available	Kernel Driver	No	Disabled
	Stopped	OK	Normal	No	No
ql12160	ql12160	Not Available	Kernel Driver	No	Disabled
	Stopped	OK	Normal	No	No
ql1240	ql1240	Not Available	Kernel Driver	No	Disabled
	Stopped	OK	Normal	No	No
ql1280	ql1280	Not Available	Kernel Driver	No	Disabled
	Stopped	OK	Normal	No	No
ql2100	ql2100	Not Available	Kernel Driver	No	Disabled
	Stopped	OK	Normal	No	No
ql2200	ql2200	Not Available	Kernel Driver	No	Disabled
	Stopped	OK	Normal	No	No
ql2300	ql2300	c:\windows.2\system32\drivers\ql2300.sys			
	Kernel Driver	Yes	Boot	Running	OK
	Normal	No	Yes		
qldirect	qldirect	c:\windows.2\system32\drivers\qldirect.sys			
	Kernel Driver	Yes	Auto	Running	OK
	Normal	No	Yes		
qlvika	qlvika	c:\windows.2\system32\drivers\qlvika.sys			
	Kernel Driver	Yes	Auto	Running	OK
	Normal	No	Yes		
rasacd	Remote Access Auto Connection Driver	c:\windows.2\system32\drivers\rasacd.sys			Kernel
Driver	Yes	System	Running	OK	Normal
	No	Yes			
rasl2tp	WAN Miniport (L2TP)	c:\windows.2\system32\drivers\rasl2tp.sys			Kernel
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
rasppoe	Remote Access PPPOE Driver	c:\windows.2\system32\drivers\rasppoe.sys			Kernel
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
raspti	Direct Parallel	c:\windows.2\system32\drivers\raspti.sys			Kernel
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
rdbs	Rdbss	c:\windows.2\system32\drivers\rdbs.sys			
	File System Driver	Yes	System	Running	
	OK	Normal	No	Yes	
rdpcdd	RDPcdd	c:\windows.2\system32\drivers\rdpcdd.sys			
	Kernel Driver	Yes	System	Running	OK
	Ignore	No	Yes		
rdpdr	Terminal Server Device Redirector Driver	c:\windows.2\system32\drivers\rdpdr.sys			Kernel
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
rdpwd	RDPWD	c:\windows.2\system32\drivers\rdpwd.sys			
	Kernel Driver	Yes	Manual	Running	OK
	Ignore	No	Yes		
redbook	Digital CD Audio Playback Filter Driver	c:\windows.2\system32\drivers\redbook.sys			Kernel
Driver	Yes	System	Running	OK	Normal
	No	Yes			
rio	Real I/O	c:\windows.2\system32\drivers\rio.sys			Kernel
Driver	No	Manual	Stopped	OK	Normal
	No	No			
serial	Serial	c:\windows.2\system32\drivers\serial.sys			
	Kernel Driver	No	Auto	Stopped	OK
	Ignore	No	No		

sfloppy	Sfloppy	c:\windows.2\system32\drivers\sfloppy.sys	Kernel Driver	No	System	Stopped	OK
	Ignore	No	No				
simbad	Simbad	Not Available	Kernel Driver	No			Disabled
	Stopped	OK	Normal	No			No
slowmemory	Slow Memory Measurement						
	c:\windows.2\system32\drivers\slowmemory.sys		Kernel				
Driver	No	Manual	Stopped	OK			Normal
	No	No					
srv	Srv	c:\windows.2\system32\drivers\srv.sys	File				
System Driver	Normal	Yes	Manual	Running	OK		
	No	Yes					
swenum	Software Bus Driver						
	c:\windows.2\system32\drivers\swenum.sys		Kernel				
Driver	Yes	Manual	Running	OK			Normal
	No	Yes					
symc8xx	symc8xx	Not Available	Kernel Driver	No			Disabled
	Stopped	OK	Normal	No			No
symmpi	symmpi	Not Available	Kernel Driver	No			Disabled
	Stopped	OK	Normal	No			No
sym_hi	sym_hi	Not Available	Kernel Driver	No			Disabled
	Stopped	OK	Normal	No			No
sym_u3	sym_u3	c:\windows.2\system32\drivers\sym_u3.sys	Kernel Driver	Yes	Boot	Running	OK
	Normal	No	Yes				
tcpip	TCP/IP Protocol Driver						
	c:\windows.2\system32\drivers\tcpip.sys		Kernel Driver	Yes			
	System	Running	OK	Normal	No		
	Yes						
tdpipe	TDPIPE	c:\windows.2\system32\drivers\tdpipe.sys	Kernel Driver	No	Manual	Stopped	OK
	Ignore	No	No				
tdtcp	TDTCP	c:\windows.2\system32\drivers\tdtcp.sys	Kernel Driver	Yes	Manual	Running	OK
	Ignore	No	Yes				
termdd	Terminal Device Driver						
	c:\windows.2\system32\drivers\termdd.sys		Kernel				
Driver	Yes	System	Running	OK			Normal
	No	Yes					
toside	Toside	Not Available	Kernel Driver	No			Disabled
	Stopped	OK	Normal	No			No
udfs	Udfs	c:\windows.2\system32\drivers\udfs.sys	File				
System Driver	Normal	No	Disabled	Stopped	OK		
	No	No					
vga	vga	c:\windows.2\system32\drivers\vgapnp.sys	Kernel Driver	No	System	Stopped	OK
	Ignore	No	No				

vgasave	VGA Display Controller.						
	c:\windows.2\system32\drivers\vgasave.sys		Kernel Driver	No			
	System	Stopped	OK	Ignore	No		
viaide	viaide	Not Available	Kernel Driver	No			Disabled
	Stopped	OK	Normal	No			No
volsnap	Storage volumes						
	c:\windows.2\system32\drivers\volsnap.sys		Kernel				
Driver	Yes	Boot	Running	OK			Normal
	No	Yes					
wanarp	Remote Access IP ARP Driver						
	c:\windows.2\system32\drivers\wanarp.sys		Kernel				
Driver	Yes	Manual	Running	OK			Normal
	No	Yes					
wdica	WDICA	Not Available	Kernel Driver	No			Manual
	Stopped	OK	Ignore	No			No
wlbs	Network Load Balancing						
	c:\windows.2\system32\drivers\wlbs.sys		Kernel Driver	No			
	Manual	Stopped	OK	Normal	No		
	No						
[Signed Drivers]							
Device Name	Signed	Device Class	Driver Version		Driver		
Date	Manufacturer	INF Name	Driver Name	Device ID			
Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	HTREE\ROOT\0				
ACPI IA64-based PC	No	COMPUTER	5.2.3783.0				
	10/1/2002	(Standard computers)	hal.inf		Not		
Available	ROOT\ACPI_HAL\0000						
Microsoft ACPI-Compliant System	No	SYSTEM					
	5.2.3783.0	10/1/2002	Microsoft	acpi.inf		Not	
Available	ACPI_HAL\PNP0C08\0						
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)		
	cpu.inf	Not Available	ACPI\GENUINEINTEL-_IA64_FAMILY_31_MODEL_1_0				
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)		
	cpu.inf	Not Available	ACPI\GENUINEINTEL-_IA64_FAMILY_31_MODEL_1_1				
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)		
	cpu.inf	Not Available	ACPI\GENUINEINTEL-_IA64_FAMILY_31_MODEL_1_2				
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)		
	cpu.inf	Not Available	ACPI\GENUINEINTEL-_IA64_FAMILY_31_MODEL_1_3				
Generic Bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)		
	machine.inf	Not Available	ACPI\HWP0001\0				
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)		
	machine.inf	Not Available	ACPI\HWP0002\0				

PCI Serial Port	Not Available	UNKNOWN	Not Available	Not			
Available	Not Available	Not Available	Not Available				
	PCI\VEN_103C&DEV_1048&SUBSYS_1223103C&REV_02\4&4F5						
EBC780800							
Ethernet Controller	Not Available	UNKNOWN	Not Available	Not			
Available	Not Available	Not Available	Not Available				
	PCI\VEN_1011&DEV_0019&SUBSYS_104F103C&REV_41\4&4F5						
EBC780808							
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)		
	machine.inf	Not Available	ACPI\HWP0002\1				
Intel(R) PRO/1000 T Server Adapter	No	NET	6.3.6.31				
	10/1/2002	Intel	nete1000.inf	Not Available			
	PCI\VEN_8086&DEV_1004&SUBSYS_10048086&REV_02\4&17A						
F9C6C&0808							
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)		
	machine.inf	Not Available	ACPI\HWP0002\2				
LSI Logic 53C1010-66 Device	No	SCSIADAPTER					
	5.2.3783.0	10/1/2002	LSI Logic Inc.				
	pnpscsi.inf	Not Available					
	PCI\VEN_1000&DEV_0021&SUBSYS_1340103C&REV_01\4&2A6						
94D11&0808							
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)		
	disk.inf	Not Available					
	SCSI\DISK&VEN_HP_36.4G&PROD_ST336706LC&REV_HP04\5&						
6459022&08000							
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)		
	disk.inf	Not Available					
	SCSI\DISK&VEN_HP_18.2G&PROD_ST318452LC&REV_HP02\5&						
6459022&08020							
CD-ROM Drive	No	CDROM	5.2.3783.0				
	10/1/2002	(Standard CD-ROM drives)	cdrom.inf		Not		
Available	SCSI\CDROM&VEN_HP&PROD_DVD-ROM_305&REV_1.01\5&6459022&08030						
4 millimeter DAT drive	No	TAPEDRIVE	5.2.3783.0				
	10/1/2002	(Standard tape drives)	tape.inf		Not		
Available	SCSI\SEQUENTIAL&VEN_HP&PROD_C5683A&REV_C111\5&645						
9022&080A0							
LSI Logic 53C1010-66 Device	No	SCSIADAPTER					
	5.2.3783.0	10/1/2002	LSI Logic Inc.				
	pnpscsi.inf	Not Available					
	PCI\VEN_1000&DEV_0021&SUBSYS_1340103C&REV_01\4&2A6						
94D11&0809							
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)		
	machine.inf	Not Available	ACPI\HWP0002\3				
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)		
	machine.inf	Not Available	ACPI\HWP0002\4				

QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.1.0 2/3/2003 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&38E
DBD97&0&08

Qlogic processor device No SYSTEM 5.2.3783.0
10/1/2002 QLOGIC scsdev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_5&29AC13B1&0&07F0
PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\5

Emulex LP9802 Fibre Channel Host Adapter No
SCSIADAPTER 5.2.3783.0 10/1/2002 Emulex
pnpscsi.inf Not Available
PCI\VEN_10DF&DEV_F980&SUBSYS_F98010DF&REV_01\4&263
40CF2&0&08

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_HP&PROD_A6218A&REV_A000\5&478A942&0
&16E0
Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_HP&PROD_A6218A&REV_A000\5&478A942&0
&16E2
PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\6

Broadcom NetXtreme Gigabit Ethernet No NET 2.91.0.0
10/1/2002 Broadcom netb57xp.inf Not Available
PCI\VEN_14E4&DEV_1645&SUBSYS_128A103C&REV_15\4&137
A5C4D&0&08

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\7

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\8

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\A

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\C

QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.1.0 2/3/2003 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&200
C63EC&0&08

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&28
20965&0&010
Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3
8\5&28202965&0&001

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&28
20965&0&010

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3
8\5&28202965&0&011

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&28
20965&0&020

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3
8\5&28202965&0&021

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.39\5&28
20965&0&030

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3
9\5&28202965&0&031

Qlogic processor device No SYSTEM 5.2.3783.0
10/1/2002 QLOGIC scsdev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_5&28202965&0&07F0

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\E

QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.1.0 2/3/2003 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&363
3CCA9&0&08

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&26
E9C916&0&000

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3
8\5&26E9C916&0&001

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&26
E9C916&0&010
Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3
8\5&26E9C916&0&011

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&26
E9C916&0&020

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3
8\5&26E9C916&0&021

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&26
E9C916&0&030

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3
8\5&26E9C916&0&031

Qlogic processor device No SYSTEM 5.2.3783.0
10/1/2002 QLOGIC scsdev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_5&26E9C916&0&07F0

ACPI Fixed Feature Button No SYSTEM 5.2.3783.0
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\FIXEDBUTTON\2&DABA3FF&0

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard
processor types) cpu.inf Not Available ACPI\GENUINEINTEL_
_IA64_FAMILY_31_MODEL_1_4

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard
processor types) cpu.inf Not Available ACPI\GENUINEINTEL_
_IA64_FAMILY_31_MODEL_1_5

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard
processor types) cpu.inf Not Available ACPI\GENUINEINTEL_
_IA64_FAMILY_31_MODEL_1_6

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard
processor types) cpu.inf Not Available ACPI\GENUINEINTEL_
_IA64_FAMILY_31_MODEL_1_7

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard
processor types) cpu.inf Not Available ACPI\GENUINEINTEL_
_IA64_FAMILY_31_MODEL_1_8

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard
processor types) cpu.inf Not Available ACPI\GENUINEINTEL_
_IA64_FAMILY_31_MODEL_1_9

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard
processor types) cpu.inf Not Available ACPI\GENUINEINTEL_
_IA64_FAMILY_31_MODEL_1_10

Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard machine.inf)
Generic Bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard machine.inf)
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard machine.inf)
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard machine.inf)
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard machine.inf)
QLogic QLA23xx PCI Fibre Channel Adapter	No				
SCSIADAPTER	8.2.1.0	2/3/2003	QLogic		
oem0.inf	Not Available				
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&3815FDFD&0&08					
Qlogic processor device	No	SYSTEM	5.2.3783.0		
10/1/2002	QLOGIC	scsdev.inf	Not Available		
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5\5&1CC61E3&0&07F0					
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard machine.inf)
Emulex LP9802 Fibre Channel Host Adapter	No				
SCSIADAPTER	5.2.3783.0	10/1/2002	Emulex		
pnpscsi.inf	Not Available				
PCI\VEN_10DF&DEV_F980&SUBSYS_F98010DF&REV_01\4&255C4D58&0&08					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
0&16C1					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
0&16C3					
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard machine.inf)
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard machine.inf)

PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard machine.inf)
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard machine.inf)
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard machine.inf)
QLogic QLA23xx PCI Fibre Channel Adapter	No				
SCSIADAPTER	8.2.1.0	2/3/2003	QLogic		
oem0.inf	Not Available				
PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&1F34A452&0&08					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002	Compaq	scsdev.inf	Not Available		
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&7680BE7&0&000					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
8\5&7680BE7&0&001					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002	Compaq	scsdev.inf	Not Available		
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&7680BE7&0&010					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
8\5&7680BE7&0&011					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002	Compaq	scsdev.inf	Not Available		
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&7680BE7&0&020					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
8\5&7680BE7&0&021					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002	Compaq	scsdev.inf	Not Available		
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&7680BE7&0&030					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
8\5&7680BE7&0&031					
Qlogic processor device	No	SYSTEM	5.2.3783.0		
10/1/2002	QLOGIC	scsdev.inf	Not Available		
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5\5&7680BE7&0&07F0					
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard machine.inf)

QLogic QLA23xx PCI Fibre Channel Adapter	No				
SCSIADAPTER	8.2.1.0	2/3/2003	QLogic		
oem0.inf	Not Available				
PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&3815FDFD&0&08					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002	Compaq	scsdev.inf	Not Available		
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&363C1E03&0&000					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
8\5&363C1E03&0&001					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002	Compaq	scsdev.inf	Not Available		
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&363C1E03&0&010					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
8\5&363C1E03&0&011					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002	Compaq	scsdev.inf	Not Available		
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&363C1E03&0&020					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
8\5&363C1E03&0&021					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002	Compaq	scsdev.inf	Not Available		
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&363C1E03&0&030					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
8\5&363C1E03&0&031					
Qlogic processor device	No	SYSTEM	5.2.3783.0		
10/1/2002	QLOGIC	scsdev.inf	Not Available		
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5\5&363C1E03&0&07F0					
Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
Generic Bus (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)

Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
		cpu.inf	Not Available	ACPI\GENUINEINTEL_-IA64_FAMILY_31_MODEL_1\16	
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
		cpu.inf	Not Available	ACPI\GENUINEINTEL_-IA64_FAMILY_31_MODEL_1\17	
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
		cpu.inf	Not Available	ACPI\GENUINEINTEL_-IA64_FAMILY_31_MODEL_1\18	
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
		cpu.inf	Not Available	ACPI\GENUINEINTEL_-IA64_FAMILY_31_MODEL_1\19	
Generic Bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
		machine.inf	Not Available	ACPI\HWP0001\4	
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
		machine.inf	Not Available	ACPI\HWP0002\40	
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
		machine.inf	Not Available	ACPI\HWP0002\41	
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
		machine.inf	Not Available	ACPI\HWP0002\42	
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
		machine.inf	Not Available	ACPI\HWP0002\43	
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
		machine.inf	Not Available	ACPI\HWP0002\44	
QLogic QLA23xx PCI Fibre Channel Adapter	No				
		SCSIADAPTER	8.2.1.0	2/3/2003	QLogic
		oem0.inf	Not Available		
		PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&A03B852&0&08			
Qlogic processor device	No	SYSTEM	5.2.3783.0		
		scsdev.inf	Not Available		
		SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5\5&90DBDA6&0&07F0			
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
		machine.inf	Not Available	ACPI\HWP0002\45	
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
		machine.inf	Not Available	ACPI\HWP0002\46	
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
		machine.inf	Not Available	ACPI\HWP0002\47	
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
		machine.inf	Not Available	ACPI\HWP0002\48	

PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
		machine.inf	Not Available	ACPI\HWP0002\4A	
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
		machine.inf	Not Available	ACPI\HWP0002\4C	
QLogic QLA23xx PCI Fibre Channel Adapter	No				
		SCSIADAPTER	8.2.1.0	2/3/2003	QLogic
		oem0.inf	Not Available		
		PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&EDA159&0&08			
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
		scsdev.inf	Not Available		
		SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&39780E2D&0&000			
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
		disk.inf	Not Available		
		SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&39780E2D&0&001			
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
		scsdev.inf	Not Available		
		SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&39780E2D&0&010			
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
		disk.inf	Not Available		
		SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&39780E2D&0&011			
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
		scsdev.inf	Not Available		
		SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&39780E2D&0&020			
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
		disk.inf	Not Available		
		SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&39780E2D&0&021			
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
		scsdev.inf	Not Available		
		SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&39780E2D&0&030			
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
		disk.inf	Not Available		
		SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&39780E2D&0&031			
Qlogic processor device	No	SYSTEM	5.2.3783.0		
		scsdev.inf	Not Available		
		SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5\5&90DBDA6&0&07F0			
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
		machine.inf	Not Available	ACPI\HWP0002\4E	

QLogic QLA23xx PCI Fibre Channel Adapter	No				
		SCSIADAPTER	8.2.1.0	2/3/2003	QLogic
		oem0.inf	Not Available		
		PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&345C0A28&0&08			
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
		scsdev.inf	Not Available		
		SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&2A293A36&0&000			
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
		disk.inf	Not Available		
		SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&2A293A36&0&001			
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
		scsdev.inf	Not Available		
		SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&2A293A36&0&010			
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
		disk.inf	Not Available		
		SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&2A293A36&0&011			
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
		scsdev.inf	Not Available		
		SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&2A293A36&0&020			
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
		disk.inf	Not Available		
		SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&2A293A36&0&021			
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
		scsdev.inf	Not Available		
		SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&2A293A36&0&030			
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
		disk.inf	Not Available		
		SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&2A293A36&0&031			
Qlogic processor device	No	SYSTEM	5.2.3783.0		
		scsdev.inf	Not Available		
		SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5\5&2A293A36&0&07F0			
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
		cpu.inf	Not Available	ACPI\GENUINEINTEL_-IA64_FAMILY_31_MODEL_1\20	
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
		cpu.inf	Not Available	ACPI\GENUINEINTEL_-IA64_FAMILY_31_MODEL_1\21	
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
		cpu.inf	Not Available	ACPI\GENUINEINTEL_-IA64_FAMILY_31_MODEL_1\22	
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
		cpu.inf	Not Available	ACPI\GENUINEINTEL_-IA64_FAMILY_31_MODEL_1\23	

Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
_IA64_FAMILY_31_MODEL_1\24					
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
_IA64_FAMILY_31_MODEL_1\25					
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
_IA64_FAMILY_31_MODEL_1\26					
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
_IA64_FAMILY_31_MODEL_1\27					
Generic Bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0001\6					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\60					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\61					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\62					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\63					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\64					
QLogic QLA23xx PCI Fibre Channel Adapter					
SCSIADAPTER			8.2.1.0	2/3/2003	QLogic
oem0.inf Not Available					
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&240E8D59&0&08					
Qlogic processor device	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard processor types)
QLOGIC scsdev.inf Not Available					
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5\5&242D741C&0&07F0					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\65					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\66					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\67					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\68					

PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\6A					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\6C					
QLogic QLA23xx PCI Fibre Channel Adapter					
SCSIADAPTER			8.2.1.0	2/3/2003	QLogic
oem0.inf Not Available					
PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&1551CFD&0&08					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard disk drives)
Compag scsdev.inf Not Available					
SCSI\ARRAY&VEN_COMPAG&PROD_MSA1000&REV_2.38\5&49CFD62&0&000					
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAG&PROD_MSA1000_VOLUME&REV_2.38\5&49CFD62&0&001					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard disk drives)
Compag scsdev.inf Not Available					
SCSI\ARRAY&VEN_COMPAG&PROD_MSA1000&REV_2.38\5&49CFD62&0&010					
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAG&PROD_MSA1000_VOLUME&REV_2.38\5&49CFD62&0&011					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard disk drives)
Compag scsdev.inf Not Available					
SCSI\ARRAY&VEN_COMPAG&PROD_MSA1000&REV_2.38\5&49CFD62&0&020					
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAG&PROD_MSA1000_VOLUME&REV_2.38\5&49CFD62&0&021					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard disk drives)
Compag scsdev.inf Not Available					
SCSI\ARRAY&VEN_COMPAG&PROD_MSA1000&REV_2.38\5&49CFD62&0&030					
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAG&PROD_MSA1000_VOLUME&REV_2.38\5&49CFD62&0&031					
Qlogic processor device	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard processor types)
QLOGIC scsdev.inf Not Available					
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5\5&49CFD62&0&07F0					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\6E					

QLogic QLA23xx PCI Fibre Channel Adapter					
SCSIADAPTER			8.2.1.0	2/3/2003	QLogic
oem0.inf Not Available					
PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&2662&0&08					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard disk drives)
Compag scsdev.inf Not Available					
SCSI\ARRAY&VEN_COMPAG&PROD_MSA1000&REV_2.38\5&B2BD377&0&000					
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAG&PROD_MSA1000_VOLUME&REV_2.38\5&B2BD377&0&001					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard disk drives)
Compag scsdev.inf Not Available					
SCSI\ARRAY&VEN_COMPAG&PROD_MSA1000&REV_2.38\5&B2BD377&0&010					
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAG&PROD_MSA1000_VOLUME&REV_2.38\5&B2BD377&0&011					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard disk drives)
Compag scsdev.inf Not Available					
SCSI\ARRAY&VEN_COMPAG&PROD_MSA1000&REV_2.38\5&B2BD377&0&020					
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAG&PROD_MSA1000_VOLUME&REV_2.38\5&B2BD377&0&021					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard disk drives)
Compag scsdev.inf Not Available					
SCSI\ARRAY&VEN_COMPAG&PROD_MSA1000&REV_2.38\5&B2BD377&0&030					
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAG&PROD_MSA1000_VOLUME&REV_2.38\5&B2BD377&0&031					
Qlogic processor device	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard processor types)
QLOGIC scsdev.inf Not Available					
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5\5&B2BD377&0&07F0					
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_1\28					
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_1\29					
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_1\30					
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_1\31					

Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard
_IA64_FAMILY_31_MODEL_1\32		cpu.inf	Not Available	ACPI\GENUINEINTEL_-	
Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard
_IA64_FAMILY_31_MODEL_1\33		cpu.inf	Not Available	ACPI\GENUINEINTEL_-	
Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard
_IA64_FAMILY_31_MODEL_1\34		cpu.inf	Not Available	ACPI\GENUINEINTEL_-	
Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard
_IA64_FAMILY_31_MODEL_1\35		cpu.inf	Not Available	ACPI\GENUINEINTEL_-	
Generic Bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard
		machine.inf	Not Available	ACPI\HWP0001\8	
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard
		machine.inf	Not Available	ACPI\HWP0002\80	
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard
		machine.inf	Not Available	ACPI\HWP0002\81	
Intel(R) PRO/1000 T Server Adapter	No	NET	6.3.6.31		
10/1/2002 Intel	nete1000.inf	Not Available			
PCI\VEN_8086&DEV_1004&SUBSYS_10048086&REV_02\4&333					
BE0D6&0&08					
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard
		machine.inf	Not Available	ACPI\HWP0002\82	
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard
		machine.inf	Not Available	ACPI\HWP0002\83	
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard
		machine.inf	Not Available	ACPI\HWP0002\84	
QLogic QLA23xx PCI Fibre Channel Adapter	No				
SCSIADAPTER	8.2.1.0	2/3/2003	QLogic		
oem0.inf	Not Available				
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&36A					
998EE&0&08					
Qlogic processor device	No	SYSTEM	5.2.3783.0		
10/1/2002 QLOGIC	scsdev.inf	Not Available			
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE					
V_5&1A52C610&0&07F0					
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard
		machine.inf	Not Available	ACPI\HWP0002\85	
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard
		machine.inf	Not Available	ACPI\HWP0002\86	
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard
		machine.inf	Not Available	ACPI\HWP0002\87	

PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard
		machine.inf	Not Available	ACPI\HWP0002\88	
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard
		machine.inf	Not Available	ACPI\HWP0002\8A	
PCI bus (system devices)	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard
		machine.inf	Not Available	ACPI\HWP0002\8C	
QLogic QLA23xx PCI Fibre Channel Adapter	No				
SCSIADAPTER	8.2.1.0	2/3/2003	QLogic		
oem0.inf	Not Available				
PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&1DC					
83F43&0&08					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002 Compaq	scsdev.inf	Not Available			
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&18					
7665B1&0&000					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard
	disk.inf	Not Available			
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3					
8\5&187665B1&0&001					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002 Compaq	scsdev.inf	Not Available			
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&18					
7665B1&0&010					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard
	disk.inf	Not Available			
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3					
8\5&187665B1&0&011					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002 Compaq	scsdev.inf	Not Available			
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&18					
7665B1&0&020					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard
	disk.inf	Not Available			
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3					
8\5&187665B1&0&021					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002 Compaq	scsdev.inf	Not Available			
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&18					
7665B1&0&030					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard
	disk.inf	Not Available			
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3					
8\5&187665B1&0&031					
Qlogic processor device	No	SYSTEM	5.2.3783.0		
10/1/2002 QLOGIC	scsdev.inf	Not Available			
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE					
V_5&1A52C610&0&07F0					
Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard
_IA64_FAMILY_31_MODEL_1\36		cpu.inf	Not Available	ACPI\GENUINEINTEL_-	
Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard
_IA64_FAMILY_31_MODEL_1\37		cpu.inf	Not Available	ACPI\GENUINEINTEL_-	
Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard
_IA64_FAMILY_31_MODEL_1\38		cpu.inf	Not Available	ACPI\GENUINEINTEL_-	
Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard
_IA64_FAMILY_31_MODEL_1\39		cpu.inf	Not Available	ACPI\GENUINEINTEL_-	

QLogic QLA23xx PCI Fibre Channel Adapter	No				
SCSIADAPTER	8.2.1.0	2/3/2003	QLogic		
oem0.inf	Not Available				
PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&7AB					
207&0&08					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002 Compaq	scsdev.inf	Not Available			
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&46					
EE788&0&000					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard
	disk.inf	Not Available			
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3					
8\5&46EE788&0&001					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002 Compaq	scsdev.inf	Not Available			
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&46					
EE788&0&010					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard
	disk.inf	Not Available			
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3					
8\5&46EE788&0&011					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002 Compaq	scsdev.inf	Not Available			
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&46					
EE788&0&020					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard
	disk.inf	Not Available			
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3					
8\5&46EE788&0&021					
Compaq MSA1000	No	SYSTEM	5.2.3783.0		
10/1/2002 Compaq	scsdev.inf	Not Available			
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&46					
EE788&0&030					
Disk drive (disk drives)	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard
	disk.inf	Not Available			
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3					
8\5&46EE788&0&031					
Qlogic processor device	No	SYSTEM	5.2.3783.0		
10/1/2002 QLOGIC	scsdev.inf	Not Available			
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE					
V_5&46EE788&0&07F0					
Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard
_IA64_FAMILY_31_MODEL_1\36		cpu.inf	Not Available	ACPI\GENUINEINTEL_-	
Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard
_IA64_FAMILY_31_MODEL_1\37		cpu.inf	Not Available	ACPI\GENUINEINTEL_-	
Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard
_IA64_FAMILY_31_MODEL_1\38		cpu.inf	Not Available	ACPI\GENUINEINTEL_-	
Processor (processor types)	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard
_IA64_FAMILY_31_MODEL_1\39		cpu.inf	Not Available	ACPI\GENUINEINTEL_-	

Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
_IA64_FAMILY_31_MODEL_1\40					
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
_IA64_FAMILY_31_MODEL_1\41					
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
_IA64_FAMILY_31_MODEL_1\42					
Processor	No	PROCESSOR	5.2.3783.0	10/1/2002	(Standard processor types)
_IA64_FAMILY_31_MODEL_1\43					
Generic Bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0001\A					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\A0					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\A1					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\A2					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\A3					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\A4					
QLogic QLA23xx PCI Fibre Channel Adapter					
SCSIADAPTER 8.2.1.0 2/3/2003 QLogic					
oem0.inf Not Available					
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&24E12DB3&0&08					
Qlogic processor device	No	SYSTEM	5.2.3783.0	10/1/2002	QLOGIC scsdev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_V_5&3497A374&0&07F0					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\A5					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\A6					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\A7					

QLogic QLA23xx PCI Fibre Channel Adapter					
SCSIADAPTER 8.2.1.0 2/3/2003 QLogic					
oem0.inf Not Available					
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&217979B3&0&08					
Qlogic processor device	No	SYSTEM	5.2.3783.0	10/1/2002	QLOGIC scsdev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_V_5&2EB863&0&07F0					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\A8					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\AA					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\AC					
QLogic QLA23xx PCI Fibre Channel Adapter					
SCSIADAPTER 8.2.1.0 2/3/2003 QLogic					
oem0.inf Not Available					
PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&227BD57&0&08					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&2A7DD39&0&000					
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&2A7DD39&0&001					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&2A7DD39&0&010					
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&2A7DD39&0&011					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&2A7DD39&0&020					
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&2A7DD39&0&021					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&2A7DD39&0&030					

Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&2A7DD39&0&010					
Qlogic processor device	No	SYSTEM	5.2.3783.0	10/1/2002	QLOGIC scsdev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_V_5&2A7DD39&0&07F0					
PCI bus	No	SYSTEM	5.2.3783.0	10/1/2002	(Standard system devices)
machine.inf Not Available ACPI\HWP0002\AE					
QLogic QLA23xx PCI Fibre Channel Adapter					
SCSIADAPTER 8.2.1.0 2/3/2003 QLogic					
oem0.inf Not Available					
PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&227B1EA1&0&08					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&33CF13EB&0&000					
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&33CF13EB&0&001					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&33CF13EB&0&010					
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&33CF13EB&0&011					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&33CF13EB&0&020					
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&33CF13EB&0&021					
Compaq MSA1000	No	SYSTEM	5.2.3783.0	10/1/2002	Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&33CF13EB&0&030					
Disk drive	No	DISKDRIVE	5.2.3783.0	10/1/2002	(Standard disk drives)
disk.inf Not Available					
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&33CF13EB&0&031					
Qlogic processor device	No	SYSTEM	5.2.3783.0	10/1/2002	QLOGIC scsdev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_V_5&33CF13EB&0&07F0					

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1144

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1145

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1146

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1147

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1148

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1149

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1150

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1151

Generic Bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0001\C

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\C0

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\C1

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\C2

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\C3

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\C4

QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.1.0 2/3/2003 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&35D6F894&0&08

Qlogic processor device No SYSTEM 5.2.3783.0
10/1/2002 QLOGIC scsdev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5&221A8769&0&07F0

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\C5

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\C6

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\C7

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\C8

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\CA

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\CC

QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.1.0 2/3/2003 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&1CF59EE9&0&08

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&F3696DA&0&000

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&F3696DA&0&001

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&F3696DA&0&010

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&F3696DA&0&011

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&F3696DA&0&020

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&F3696DA&0&021

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&F3696DA&0&030

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&F3696DA&0&000

Qlogic processor device No SYSTEM 5.2.3783.0
10/1/2002 QLOGIC scsdev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5&F3696DA&0&07F0

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\CE

QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.1.0 2/3/2003 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&8331D07A6&0&08

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&36CA00&0&000

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&36CA00&0&001

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&36CA00&0&010

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&36CA00&0&011

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&36CA00&0&020

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&36CA00&0&021

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&36CA00&0&030

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&36CA00&0&031

Qlogic processor device No SYSTEM 5.2.3783.0
10/1/2002 QLOGIC scsdev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5&3636CA00&0&07F0

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1\52

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1\53

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1\54

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1\55

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1\56

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1\57

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1\58

Processor No PROCESSOR 5.2.3783.0 10/1/2002 (Standard processor types) cpu.inf Not Available ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1\59

Generic Bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0001\E

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\E0

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\E1

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\E2

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\E3

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\E4

QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.1.0 2/3/2003 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&7C4B2E9&0&08

Qlogic processor device No SYSTEM 5.2.3783.0
10/1/2002 QLOGIC scsdev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5\5&386DAF6B&0&07F0

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\E5

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\E6

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\E7

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\E8

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\EA

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\EC

QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.1.0 2/3/2003 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&11CA6C2&0&08

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&2A6661DB&0&000

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&2A6661DB&0&001

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&2A6661DB&0&010

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&2A6661DB&0&011

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&2A6661DB&0&020

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&2A6661DB&0&021

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&2A6661DB&0&030

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&2A6661DB&0&031

Disk drive No SYSTEM 5.2.3783.0
10/1/2002 QLOGIC scsdev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5\5&2A6661DB&0&07F0

PCI bus No SYSTEM 5.2.3783.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\EE

QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.1.0 2/3/2003 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01000E11&REV_02\4&3690080C&0&08

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&18FC85D9&0&000

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&18FC85D9&0&001

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&18FC85D9&0&010

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&18FC85D9&0&011

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&18FC85D9&0&020

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&18FC85D9&0&021

Compaq MSA1000 No SYSTEM 5.2.3783.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\5&18FC85D9&0&030

Disk drive No DISKDRIVE 5.2.3783.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\5&18FC85D9&0&031

Qlogic processor device No SYSTEM 5.2.3783.0
10/1/2002 QLOGIC scsdev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5\5&18FC85D9&0&07F0

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BCD3A33B-6A76-44ED-932D-D6D08F5C492E}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{42933E11-E757-4F7A-8997-409B658124B8}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EB418173-8B46-4067-8088-7DB8497DAE07}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3EA7F9ED-CAE7-476D-818A-3F6FF3B4235}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{046F78A5-FB37-47C5-87AA-691842BBE8AE}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FB1B2BE1-DFB4-42D3-98AA-0237A6C2E0BA}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CC5AD891-25BE-4F74-903C-AE4B20ED6B53}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{501205F9-22E7-48B2-BDDE-B693D7FD37AB}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6CA12C8F-286C-11D7-9D9F-505054503030}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6CA12C93-286C-11D7-9D9F-505054503030}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{66E5AB59-F087-47FA-A5BC-1746838D1A94}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3D0636E7-0BD4-4458-A817-098805130014}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{24E0E69F-EE24-43F3-B843-130D781D9039}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3F822E8C-9A73-436A-9E5-4E8F5F27AEE}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5FB7B252-DC1C-4EEF-A38C-FFDD2AE57AFD}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{123235BA-D799-4482-AD9E-A8E043EB8DF0}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{192EDB78-99C0-44B0-A62E-452F6E582464}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6A5DD2A-5C72-4247-B000-A91142848D4C}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{252F068E-513B-46A7-963C-A0F0AB44EABB}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E60D518D-3080-4C21-B4D2-519F49222BC6}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D93A478C-1949-41CF-BA95-B9F767E28546}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9CCC7383-05D4-4196-8BA2-08FF175CA553}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6433E983-72D9-4A19-8D03-CDB083C73400}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4822628E-4929-42D0-BBD7-F8AABC97B301}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5070958D-D016-438C-AF25-CC0F6A66D1FF}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8E670E5F-41AE-418A-941D-089441F4EE90}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B1909DA5-6978-4C9D-AE79-EFA953C19E80}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E7B9E973-79E2-41F2-B2AF-019EAB91FC40}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{300E1CFD-4617-40CF-AE8E-82DECDFD8B42D}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F3270DF8-5EB3-4956-BF4D-CC35A3FCA4CD}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{54907F3D-19F5-4E8E-A057-0A651E45F157}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E46A41CD-02F9-4EC7-9AC6-AA78159B6DA5}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EFC7A250-B771-4ACF-A599-61E52D2EC2BE}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2514769A-5BB8-4E00-A594-E0DA04EE51FE}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{80DE6847-87A1-4628-BCD1-8C336A9B25D9}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EF5B6A38-89E1-42E0-B589-21595A3BF21E}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0AABCB37-8AEF-4A4E-A919-D2E3030E78D2}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{41FCD3D9-324E-41E7-8CD3-B1C14E8A12E9}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8A5FF5A6-2EDF-49F4-8941-AE58572844F3}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{54E74DAB-3A86-47E1-AAE9-5CD267051FB9}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4565668A-8C8B-47F6-851A-51BFEEBBF800}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1EB109C1-6A0E-42D4-ABC2-C5E98483D1DE}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{91008745-7E47-4FE1-8471-86C8C6C1F8F3}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7A02BE82-2D8F-4929-859A-033578944DBC}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0C2C5810-C377-4112-A2FE-4756AA7B45B3}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BBE1F31E-7C67-4045-8D7C-7AE35FEF4A88}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8BDEFB37-BA89-4B67-A183-EBB05C400589}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{053A190E-D850-4699-AD98-5E1B2B29482A}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9F614E88-0ECF-495A-BFCE-2EA91EE7BBE1}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A0CFFE68-567D-4C24-AF83-C1D016250517}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F9DC42F-155A-4F06-9FC5-1FCD3576D11B}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0E67EDE8-092B-4FBC-A13D-7FD3E4C2B753}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{108302EC-394F-4693-87E1-5A602AD9D612}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A4DD025-4-8D0C-4DEF-AB58-F04B0C53B90D}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{73DFD8B4-7194-4753-AC24-8BD7B21B191E}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{318648CB-E33B-4C26-974C-B99BB8075455}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C94898DF-92F1-4CFD-9162-70977D7DF992}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9F7A55F7-E0D3-44A0-8A15-FA8405B8DD3D}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{211BA87A-8C73-4659-959C-C74E436798A6}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B1179872-B6EE-43D4-BEB0-7DE92D9858BC}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D9C1D4A-1-4256-435B-B0A5-408D0E1590B1}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{17ED6111-2D1F-4C59-8D37-9E049021F883}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{60166C78-ABD2-4E2F-84A8-D549A31B589A}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9C8E2E78-D7A7-4E61-AF45-0F134A213A06}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C784E6E5-65BE-42F7-847C-74FCB66D216E}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D586A03A-98E1-4C0A-80C4-FE66844B9C79}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1A187A93-0F6B-452F-8F75-029DC80DA8F7}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8460B88F-A871-41B0-B7D4-D89637502806}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{579C6A10-C047-45A0-8BFF-4043050A8714}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{737F82A5-64E0-4C88-B7D2-816B53F3259E}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{129B0F68-FB55-4027-9F89-A706FFF328A1}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1EDDD9A-F-E33E-4AF4-809C-63168F974C10}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E1376F2E-B812-4DFD-83EE-811D33181BC5}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{525EA8AA-D43B-4FAF-80EA-8763FDD2AA1D}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5307CEDA-4732-4CF2-9FEE-A2380B7B5D61}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{57BA9C6C-9842-4A53-91CE-004B1871FA19}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{05F62240-84BF-406A-9F2D-1BC5EEABC7F9}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{12040552-1650-4A31-9D5E-BE8D324542BC}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{837B3F97-1301-437D-B6D3-4B731F614CBB}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FD2E2C14-C7FA-4EFE-A66C-9A81B3B007E0}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{689AA87F-6E48-465A-915F-5F1F81E24B7F}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C98DF1D8-BF40-4881-B427-69A8CD47E936}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{04A11771-F0A0-496E-ACC0-746A0449B39C}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{95A21C96-6DB1-4DB6-8F50-61AB3E731475}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0C0024B8-B2E6-402E-8732-05D89088777F}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{82EEA449-6EFB-4432-9EB5-1077D553E16E}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{05CC389A-3DBE-434B-B61A-3EE0D21EB3CB}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7A1D805B-E601-4A23-8A37-E6D4D4E233CD}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6C5BDF3E-A9A1-4D8A-89FC-486A7FA39882}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F893D439-35A8-4BD5-BCAF-6EA0B723B401}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3A08547E-A656-4987-9101-956593A893A8}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{839EBDE0-9A52-40E1-8223-B236A8C77F14}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8E6187DD-CB26-40AE-BDE4-720A675CF40E}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FC9CB054-8C31-4C66-BE71-0243F0C31F6F}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D1C15B26-F86A-4146-A937-D88BE5883DDA}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3138A765-709D-4709-ABE7-EDC7E71012BF}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1B3BF7A4-E6A5-4362-B652-FFB7AF3B5765}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EEA541A8-9296-4CC6-992E-DB8586EF3C12}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EFBE08D9-4A53-41A0-8DB6-195B6AA7FE1B}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FBAEB896-E133-47A9-94DD-4CF4B01C0F0A}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{ACB85281-C10E-4AC6-A16D-D46CCDAC2FE6}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{01AD6EE5-2189-41E4-B176-A86508EA8516}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8FFD263D-0511-4530-BF3B-6E84792E5517}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A2BB5A87-379E-4742-8E69-1DC88982F73}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6AB82B4F-9748-4294-B446-63424AAB923C8}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{40625935-EF28-44B0-8292-A60DBC8241B3}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FB5844BE-CF69-4ECE-A37E-7BC1AA9B4FB1}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1C23F668-1E3F-45FC-88E4-D337C687457E}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8A54B348-E2B2-43B0-8E3D-B7F3BF0B91FE}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{660D1632-9BAF-4359-A851-1149E86B508B}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{97B99E12-2460-44D9-9197-5C2FB4838B57}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9FB756F5-DDFB-4D5D-90D3-92E921A561CF}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9037270A-6A52-4989-B0D1-479B4E6346C5}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DDAB0A46-233A-407A-A5C1-63A1A4B3B876}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6A68D9D6-C2F3-4D61-A57B-B5538322D3E8}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{86EB0F6C-6B3F-48A7-8FE5-DB724942789A}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E5FBD627-A1CF-438C-A54D-A56DDDD009F}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{91178410-0325-497A-A749-29D64003D2C5}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BFE6A1A6-305E-4C9D-B2AD-73DC36D0B8BA}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9494FFB9-1237-4973-8379-EF1ACEA737B7}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B8BB94E4-A43C-4114-A94D-B9F84C5A5000}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5A700F1A-8CAD-43AE-8326-1E4710407BEA}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8D60E035-6717-452C-8B0C-A999C317F5E3}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BC0A378F-1975-46F2-BACF-47CD98FE279}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9AA7EDA E-47BF-4352-AECF-D6C28B0A54C0}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F74FEF21-C928-4CB4-BD4E-F069F594FFF2}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1EAB8C48-CD05-47A4-96B8-6D3EBD027131}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3C5BD7B C-E111-487D-A612-70DD468F47E}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0FD10F86-5E0B-4359-B3AB-2B2ED763C011}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{529F03D6-13FF-4690-AD01-A137B67B5895}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B7EDA48F-8284-4501-8043-B0E356D8EA97}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C0538852-C9DD-4EA2-B35E-5D13F4739FBB}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{94BF2ACD-857E-4520-8FE1-4FD1AF79525C}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6729AE05-201F-4E94-A83D-FOEB6ED51765}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CA66BDB 4-1CC5-40AD-8756-0B8DB28DF47}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{624B81C4-EA2F-4C2E-95B5-1B652FDDA55C}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{392C5913-6FBE-439F-8EC7-4AA082AE2374}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0AB3AABC-ACEB-4D7B-ADB5-E7CB934CEF6C}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C0E5DBF9-66B3-4D1D-8F62-356D1D154DEB}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F5AC62CA-A2C7-4E22-8876-DB88F430E51}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F9EB228D-8A8E-4E03-9DD1-73385D30D762}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BBB51302-7D81-4B33-A5C5-576F27268C09}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CB730791-C612-4498-AAEC-0C1472DCDF93}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4B0017A9-9A74-4501-8043-B0E356D8EA97}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{01F4071F-2333-4721-A34D-DD649339D19B}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CA8BF999-AB4E-483E-A526-BEB16C804758}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5E7EEF1F-01B7-4693-9986-BC1C4EEB704E}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A6D21091-BA03-4A2E-A333-B5E9A16E6C28}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D4E82BBC-49CA-4B4E-9B48-70D15604334C}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{97B96D04-7F2C-4CB4-A4DE-2B9703082ABC}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{38B93BD2-4C51-4FBF-B0D9-2F0D36D70F41}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DA19D2F B-D15C-4377-A5EB-DE4ECE3292F4}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1E146289-A25D-4C2A-9AEC-3BF091975E77}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3455F5B6-0667-4F13-98D7-2305EF52E6FA}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6DB7D3A 6-D7E7-4957-AB9F-DC8C9F72E691}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{023C420E-8253-45F0-8D72-9D4F5AD015B}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AA80C5E1-405C-4425-957D-A3878CEDD0EC}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{29616BD9-116C-4C28-889A-9842FAD1E08E}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B7E20265-E860-47C3-872A-988F24E2BEA6}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7935C058-944F-485E-9D6F-B3BD680E22F9}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BBC009C6-ADFD-4DB1-BCF5-56ED862575B7}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0AB99D28-D7DA-4343-A895-E7BBA4E3BDF0}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D4342D8D-32F8-46C1-B827-38FA45402AD7}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A7A275E9-5DBB-409F-8CDA-CD4450706F56}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E0070127-DB7B-4736-9F95-FE87415CB951}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{705D66B0-B0B3-40AB-9B2B-C8021BC34273}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{48B8190A-8F10-4E4D-81AE-9A74ADE6C014}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{345A829E-CE7B-47DA-87CA-D15D8696F15A}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A18225C4-3732-43B5-A6E5-62CB40F2341B}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DA4D861-405C-4425-957D-A3878CEDD0EC}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A0C370E3-9187-4154-92B7-F1D6B4906E5B}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F36CE964-CB17-4260-B48F-6376F09A34EA}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{677F42E2-7DC2-45C2-88D4-7DEDAD8BD91D}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B600A618-5E6C-40AA-BC3E-AEA33FD91508}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A39B9810-5342-4253-9FCE-98268CEBB050}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{66D58DA-A-D1CB-43B6-ACCD-97D7FBA4D20C}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{422411C4-1348-490D-8B31-227E064CEDBE}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5D5B8174-1449-4F5B-8AD9-54DFA34EFDEA}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B47B01AA-5783-4DA9-A37D-F98D63A1AD35}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{577067D3-3D84-49AC-80C3-6D35CC53E3DE}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5D07FC84-93DB-4B94-803C-B72481E53695}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E7C492A3-7C45-4446-8312-DAFCDBD3711D}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2FD3A480-405C-4425-957D-A3878CEDD0EC}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0E318967-B252-4DB0-9190-5A3973C1F599}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{10C2E272-46D6-4A3A-B289-3C8CD66103A8}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6511D8AC-E9D2-43C8-83C6-9192403D3305}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A809FD82-43D5-41AB-A163-B70762643A7F}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9A90D38B-D2BD-4A18-BC53-E48C959E8A00}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DF364A94-6964-46E4-91D9-21F96C95E3BC}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F37CF1E2-3DD9-42CE-B7D8-5811293777F2}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5B5CFFFB-9A81-4667-BEDF-10008AAF7EED}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{43E4BBE6-DE1E-4982-B911-9DEC8FBAEB80}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{74BF29C2-4CBA-420B-8315-405600F8F5CF}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E0ABD324-B0B4-4A48-81FB-19C1484EEFF1}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{78C47941-1233-40BF-B073-3858FD3CD334}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B643B8A6-E7BB-4207-AAFF-340CBD21F01B}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{37AFE76-B072-4CBD-8CC1-F331DF7DEBDE}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{495D8561-3381-43D3-9C4B-33D2CA376FD6}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6E48FBAB-3EDB-4572-B1E7-7137A0353EC2}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E90BA68B-9835-4E0E-BD5A-9F9885FD656A}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{60828DEA-59C3-4B71-B8F9-625A42F4C944}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5DEB59F3-C369-4402-ADF3-B34CC51A3F0C}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1869BBF2-FF6B-4ED3-AB3A-090D28FD4CB7}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6F46F671-A773-4A99-B56A-6E734F24A008}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{944AABBF-2EEA-420D-9DA2-168ABF3D4894}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4FC227E1-BECF-49C9-82B7-F3091734E847}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{ADEE4763-CE89-4228-9B21-034273E38533}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{35055B7C-E86A-4CB9-94EA-3E581504F6A0}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{783F5C5C-9984-70B8-7F7-6BA01C3AF2AC}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{982ADC79-CE06-4A28-9BCC-DC609EE011FF}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{93742EF6-9DC8-46E1-8380-BB3A7ACBF1FD}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9175A206-DE66-41FE-B47F-E37A1C7C9EC8}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C9E7BB19-E8B8-4677-8C4B-EB9DABD91CB1}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FC9892DE-07C0-4317-8A4D-9D6846057C1D}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EC79E1C4-6C51-4696-BA39-3EF9AD3491F6}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B948AA4C-7490-4B92-9A35-56B021E2457C}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{43A8CFCC-E804-436A-A2D8-D45B474ED049}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{858ECBE2-6C4D-4F3C-9ADB-734FF645CAE4}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3E6D0D79-3174-442B-BFF4-D1B6F24A6B2E}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{07D52CD-E9B27-459D-AD9F-6AE14A81D7E2}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{05C8184B-A2BE-4514-BF8B-7B34D2178113}			

Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B649A4DA-9984-70B8-7F7-6BA01C3AF2AC}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F80AAA0B-B073-48CF-98B8-C4F991A71A19}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{567E1D81-BF23-4570-8943-76E4E90688B5}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{26EC0FE4-DA47-4F42-8EA6-320B53622FE5}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4BC70AEC-1CB7-4651-986B-D50FD2257657}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{404514B9-AE64-4505-97FF-47A309F4C1FA}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{17E85198-5EE7-4497-99DE-E1F6E9FF5FE5}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{88BDB13A-7B62-4147-8118-4F83580F0A74}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{02B0AFEB-D7EA-408F-A9CD-26DAC713C242}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1E612E23-720D-4361-B57D-4743ADF01598}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EDE96788-C7CB-4CB4-8267-548181AAB690}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5A52E959-7F32-4761-9EEA-A1F900DFDFE7}			
Generic volume	No	VOLUME	5.2.3783.0
10/1/2002 Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7EF0C818-1403-4127-BBE9-76C6D8AB8BFC}			

AFD Networking Support Environment	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
Beep	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
CRC Disk Filter Driver	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
dmbboot	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
dmload	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
Fips	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
Generic Packet Classifier	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
IPSEC driver	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
ksecdd	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
mountmgr	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
NDIS System Driver	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
Remote Access NDIS TAPI Driver	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
NDIS Usermode I/O Protocol	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
NDProxy	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
NetBios over Tcpip	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
Null	Not Available	LEGACYDRIVER
Available	Not Available	Not Available
Partition Manager	Not Available	LEGACYDRIVER
Available	Not Available	Not Available

ProcInfo	Not Available	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not Available	Not
qldirect	Not Available	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not Available	Not
qlvika	Not Available	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not Available	Not
Remote Access Auto Connection Driver	Not Available	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not Available	Not
RDPCDD	Not Available	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not Available	Not
RDPWD	Not Available	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not Available	Not
sacdrv	Not Available	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not Available	Not
Slow Memory Measurement	Not Available	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not Available	Not
TCP/IP Protocol Driver	Not Available	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not Available	Not
TDTCP	Not Available	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not Available	Not
volsnap	Not Available	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not Available	Not
Remote Access IP ARP Driver	Not Available	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not Available	Not
Audio Codecs	No	MEDIA	5.2.3783.0	10/1/2002 (Standard system devices)
Legacy Audio Drivers	No	MEDIA	5.2.3783.0	10/1/2002 (Standard system devices)
Available	Not Available	Not Available	Not Available	Not
Media Control Devices	No	MEDIA	5.2.3783.0	10/1/2002 (Standard system devices)
Available	Not Available	Not Available	Not Available	Not
Legacy Video Capture Devices	No	MEDIA	5.2.3783.0	10/1/2002 (Standard system devices)
Video Codecs	No	MEDIA	5.2.3783.0	10/1/2002 (Standard system devices)
WAN Miniport (L2TP)	No	NET	5.2.3783.0	10/1/2002 Microsoft netrasa.inf

WAN Miniport (IP)	No	NET	5.2.3783.0	10/1/2002 Microsoft netrasa.inf
WAN Miniport (PPPoE)	No	NET	5.2.3783.0	10/1/2002 Microsoft netrasa.inf
WAN Miniport (PPTP)	No	NET	5.2.3783.0	10/1/2002 Microsoft netrasa.inf
Direct Parallel	No	NET	5.2.3783.0	10/1/2002 Microsoft netrasa.inf
Terminal Server Device Redirector	No	SYSTEM	5.2.3783.0	10/1/2002 (Standard system devices)
Terminal Server Keyboard Driver	No	SYSTEM	5.2.3783.0	10/1/2002 (Standard system devices)
Terminal Server Mouse Driver	No	SYSTEM	5.2.3783.0	10/1/2002 (Standard system devices)
Plug and Play Software Device Enumerator	No	SYSTEM	5.2.3783.0	10/1/2002 (Standard system devices)

[Environment Variables]

Variable	Value	User Name
ClusterLog	C:\WINDOWS.2\Cluster\cluster.log	<SYSTEM>
ComSpec	%SystemRoot%\system32\cmd.exe	<SYSTEM>
NUMBER_OF_PROCESSORS	64	<SYSTEM>
OS	Windows_NT	<SYSTEM>
Path	%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\Program Files\Microsoft SQL Server\80\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\80\Tools\Binn\;C:\Program Files\Debugging Tools for Windows 64-bit;c:\jdw	<SYSTEM>
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH	<SYSTEM>
PROCESSOR_ARCHITECTURE	IA64	<SYSTEM>
PROCESSOR_IDENTIFIER	ia64 Family 31 Model 1 Stepping 5, GenuineIntel	<SYSTEM>
PROCESSOR_LEVEL	31	<SYSTEM>
PROCESSOR_REVISION	0105	<SYSTEM>
TEMP	%SystemRoot%\TEMP	<SYSTEM>
TMP	%SystemRoot%\TEMP	<SYSTEM>
windir	%SystemRoot%	<SYSTEM>
TEMP	%USERPROFILE%\Local Settings\Temp	NT
AUTHORITY\SYSTEM	%USERPROFILE%\Local Settings\Temp	NT
AUTHORITY\SYSTEM	%USERPROFILE%\Local Settings\Temp	NT
TEMP	%USERPROFILE%\Local Settings\Temp	NT
SERVICE	NT AUTHORITY\LOCAL	

TMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\LOCAL SERVICE

TEMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\NETWORK SERVICE

TMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\NETWORK SERVICE

_NT_SYMBOL_PATH c:\program files\microsoft sql server\mssql\bin\;s:\misc;s:\dnplus.bbt;s:\srv03.qfe.bbt;s:\dnplus.ExtInts;s:\dnplus.c2;s:\dnplus.tf;s:\dnplus.3790.apcfastpath

SQLORCA\Administrator

TEMP %USERPROFILE%\Local Settings\Temp SQLORCA\Administrator

TMP %USERPROFILE%\Local Settings\Temp SQLORCA\Administrator

[Print Jobs]

Document Submitted	Size	Owner	Notify	Status	Time
Job ID	Priority	Parameters	Driver	Print	
Processor	Host Print Queue	Data Type	Name		

[Network Connections]

Local Name	Remote Name	Type	Status	User
Z:	\\ssd\\lab\d\$	Disk	Persistent Connection	
	SQLORCA\Administrator			

[Running Tasks]

Name	Path	Process ID	Priority	Min Working Set
Max Working Set	File Date	Start Time	Version	Size
system idle process	Not Available	0	0	Not Available
Available	Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	Not Available	Not Available
system	Not Available	4	8	0
smss.exe	Not Available	592	11	409600
Available	4/22/2003 9:13 AM	Not Available	Not Available	Not Available
csrss.exe	Not Available	656	13	Not Available
Available	4/22/2003 9:13 AM	Not Available	Not Available	Not Available
Available	Not Available	656	13	Not Available
winlogon.exe	c:\windows.2\system32\winlogon.exe	680	13	409600
bytes)	5.2.3783.0 (srv03_rtm.030305-2003)	618.00 KB (632,832)	3/7/2003 9:11 AM	

services.exe	c:\windows.2\system32\services.exe	724	9	409600	2826240	4/22/2003 9:13 AM
bytes)	5.2.3783.0 (srv03_rtm.030305-2003)	285.50 KB (292,352)	3/7/2003 9:11 AM			
lsass.exe	c:\windows.2\system32\lsass.exe	736	9	409600	2826240	4/22/2003 9:13 AM
bytes)	5.2.3783.0 (srv03_rtm.030305-2003)	15.00 KB (15,360)	3/7/2003 9:10 AM			
svchost.exe	c:\windows.2\system32\svchost.exe	920	8	409600	2826240	4/22/2003 9:14 AM
bytes)	5.2.3783.0 (srv03_rtm.030305-2003)	32.50 KB (33,280)	3/7/2003 9:11 AM			
svchost.exe	c:\windows.2\system32\svchost.exe	956	8	409600	2826240	4/22/2003 9:14 AM
bytes)	5.2.3783.0 (srv03_rtm.030305-2003)	32.50 KB (33,280)	3/7/2003 9:11 AM			
svchost.exe	Not Available	1128	8	Not Available	Not Available	Not Available
Available	4/22/2003 9:14 AM	Not Available	Not Available	Not Available	Not Available	Not Available
svchost.exe	Not Available	1208	8	Not Available	Not Available	Not Available
Available	4/22/2003 9:14 AM	Not Available	Not Available	Not Available	Not Available	Not Available
svchost.exe	c:\windows.2\system32\svchost.exe	1276	8	409600	2826240	4/22/2003 9:14 AM
bytes)	5.2.3783.0 (srv03_rtm.030305-2003)	32.50 KB (33,280)	3/7/2003 9:11 AM			
msdtc.exe	Not Available	1360	8	Not Available	Not Available	Not Available
Available	4/22/2003 9:14 AM	Not Available	Not Available	Not Available	Not Available	Not Available
csrss.exe	Not Available	1176	13	Not Available	Not Available	Not Available
Available	4/22/2003 9:14 AM	Not Available	Not Available	Not Available	Not Available	Not Available
winlogon.exe	c:\windows.2\system32\winlogon.exe	1232	13	409600	2826240	4/22/2003 9:14 AM
bytes)	5.2.3783.0 (srv03_rtm.030305-2003)	618.00 KB (632,832)	3/7/2003 9:11 AM			
dfssvc.exe	c:\windows.2\system32\dfssvc.exe	2508	8	409600	2826240	4/22/2003 9:14 AM
bytes)	5.2.3783.0 (srv03_rtm.030305-2003)	441.50 KB (452,096)	3/7/2003 9:10 AM			
rdpclip.exe	c:\windows.2\system32\rdpclip.exe	2628	8	409600	2826240	4/22/2003 9:15 AM
bytes)	5.2.3783.0 (srv03_rtm.030305-2003)	156.00 KB (159,744)	3/7/2003 9:40 AM			
explorer.exe	c:\windows.2\explorer.exe	2712	8	409600	2826240	4/22/2003 9:15 AM
(srv03_rtm.030305-2003)	1.63 MB (1,704,960 bytes)	3/7/2003 9:10 AM				
wmiprvse.exe	Not Available	2940	8	Not Available	Not Available	Not Available
Available	Not Available	4/22/2003 9:15 AM	Not Available	Not Available	Not Available	Not Available
Available	Not Available					

cmd.exe	c:\windows.2\system32\cmd.exe	3224	8	409600	2826240	4/22/2003 9:15 AM
bytes)	5.2.3783.0 (srv03_rtm.030305-2003)	493.50 KB (505,344)	3/7/2003 9:10 AM			
sqlservr.exe	c:\windows.2\system32\sqlservr.exe	3264	13	409600	2826240	4/22/2003 9:15 AM
(24,153,600 bytes)	2000.080.0782.00	23.03 MB	4/20/2003 8:26 AM			
cmd.exe	c:\windows.2\system32\cmd.exe	3736	8	409600	2826240	4/22/2003 9:17 AM
bytes)	5.2.3783.0 (srv03_rtm.030305-2003)	493.50 KB (505,344)	3/7/2003 9:10 AM			
helpctr.exe	c:\windows.2\pchealth\helpctr\binaries\helpctr.exe	3760	8	409600	2826240	4/22/2003 9:17 AM
bytes)	5.2.3783.0 (srv03_rtm.030305-2003)	1.97 MB (2,066,432)	3/7/2003 9:42 AM			
wmiprvse.exe	Not Available	3820	8	Not Available	Not Available	Not Available
Available	Not Available	4/22/2003 9:17 AM	Not Available	Not Available	Not Available	Not Available
Available	Not Available					
helpsvc.exe	c:\windows.2\pchealth\helpctr\binaries\helpsvc.exe	3844	8	409600	2826240	4/22/2003 9:17 AM
bytes)	5.2.3783.0 (srv03_rtm.030305-2003)	2.18 MB (2,289,152)	3/7/2003 9:42 AM			

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer Path
winlogon	5.2.3783.0 (srv03_rtm.030305-2003)	618.00 KB (632,832 bytes)	3/7/2003 9:11 AM	Microsoft Corporation
ntdll	5.2.3775.0 (dnsrv_dev(v-thief).030303-0348)	1.45 MB (1,524,736 bytes)	3/18/2003 9:41 AM	Microsoft Corporation
kernel32	5.2.3790.2 (dnsrv_dev(v-thief).030401-1637)	1.76 MB (1,850,368 bytes)	4/8/2003 8:27 PM	Microsoft Corporation
msvcrt	7.0.3783.0 (srv03_rtm.030305-2003)	873.50 KB (894,464 bytes)	3/7/2003 9:10 AM	Microsoft Corporation
advapi32	5.2.3783.0 (srv03_rtm.030305-2003)	1.32 MB (1,383,424 bytes)	3/7/2003 9:10 AM	Microsoft Corporation
rpcrt4	5.2.3783.0 (srv03_rtm.030305-2003)	2.02 MB (2,116,096 bytes)	3/7/2003 9:11 AM	Microsoft Corporation
user32	5.2.3783.0 (srv03_rtm.030305-2003)	1.31 MB (1,372,672 bytes)	3/7/2003 9:11 AM	Microsoft Corporation
gdi32	5.2.3783.0 (srv03_rtm.030305-2003)	783.00 KB (801,792 bytes)	3/7/2003 9:10 AM	Microsoft Corporation

userenv bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\userenv.dll	1.46 MB (1,536,000 bytes)
nddeapi bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\nddeapi.dll	39.50 KB (40,448 bytes)
crypt32 bytes)	5.131.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\crypt32.dll	1.50 MB (1,576,448 bytes)
msasn1 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\msasn1.dll	153.50 KB (157,184 bytes)
secur32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\secur32.dll	167.00 KB (171,008 bytes)
winsta bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\winsta.dll	138.50 KB (141,824 bytes)
netapi32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\netapi32.dll	832.00 KB (851,968 bytes)
profmap bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\profmap.dll	55.50 KB (56,832 bytes)
regapi bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\regapi.dll	124.00 KB (126,976 bytes)
ws2_32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\ws2_32.dll	228.50 KB (233,984 bytes)
ws2help bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\ws2help.dll	49.50 KB (50,688 bytes)
msgina bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\msgina.dll	1.35 MB (1,417,728 bytes)
shsvcs bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\shsvcs.dll	321.50 KB (329,216 bytes)
shlwapi bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\shlwapi.dll	722.00 KB (739,328 bytes)
sfc bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\sfc.dll	7.50 KB (7,680 bytes)
sfc_os bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\sfc_os.dll	257.00 KB (263,168 bytes)
wintrust bytes)	5.131.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\wintrust.dll	451.50 KB (462,336 bytes)

ole32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\ole32.dll	3.38 MB (3,549,184 bytes)
imagehlp bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\imagehlp.dll	128.50 KB (131,584 bytes)
comctl32 bytes)	6.0 (srv03_rtm.030305-2003) 3/7/2003 9:20 AM Microsoft Corporation c:\windows.2\winsxs\ia64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.100.0_x-ww_b3722bab\comctl32.dll	2.18 MB (2,285,056 bytes)
version bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\version.dll	44.00 KB (45,056 bytes)
sxs bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\sxs.dll	1.77 MB (1,860,608 bytes)
winscard bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\winscard.dll	291.50 KB (298,496 bytes)
wtsapi32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\wtsapi32.dll	47.50 KB (48,640 bytes)
winmm bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\winmm.dll	404.00 KB (413,696 bytes)
shell32 bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\shell32.dll	12.35 MB (12,953,088 bytes)
setupapi bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\setupapi.dll	1.90 MB (1,991,168 bytes)
wldap32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\wldap32.dll	413.50 KB (423,424 bytes)
cscdll bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\cscdll.dll	211.00 KB (216,064 bytes)
wlnotify bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\wlnotify.dll	218.00 KB (223,232 bytes)
winspool bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\winspool.drv	399.50 KB (409,088 bytes)
mpr bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:20 AM Microsoft Corporation c:\windows.2\system32\mpr.dll	163.00 KB (166,912 bytes)
comctl32 bytes)	5.82 (srv03_rtm.030305-2003) 3/7/2003 9:20 AM Microsoft Corporation c:\windows.2\winsxs\ia64_microsoft.windows.common-controls_6595b64144ccf1df_5.82.0.0_x-ww_b9c4a0a5\comctl32.dll	1.55 MB (1,621,504 bytes)
uxtheme bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\uxtheme.dll	527.50 KB (540,160 bytes)

services bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\services.exe	285.50 KB (292,352 bytes)
scesrv bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\scesrv.dll	765.00 KB (783,360 bytes)
authz bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\authz.dll	202.50 KB (207,360 bytes)
umpnpgm bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\umpnpgm.dll	314.50 KB (322,048 bytes)
ncobjapi bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\ncobjapi.dll	118.50 KB (121,344 bytes)
msvcp60 bytes)	6.10.2240.8 941.50 KB (964,096 bytes) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\msvcp60.dll	
eventlog bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\eventlog.dll	157.00 KB (160,768 bytes)
psapi bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\psapi.dll	48.00 KB (49,152 bytes)
lsass bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\lsass.exe	15.00 KB (15,360 bytes)
lsasrv bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\lsasrv.dll	1.94 MB (2,034,688 bytes)
samsrv (1,041,920 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\samsrv.dll	1,017.50 KB
cryptdll bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\cryptdll.dll	61.00 KB (62,464 bytes)
dnsapi bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\dnsapi.dll	404.00 KB (413,696 bytes)
samlib bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\samlib.dll	106.00 KB (108,544 bytes)
ntdsapi bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\ntdsapi.dll	182.00 KB (186,368 bytes)
msprivs bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\msprivs.dll	46.00 KB (47,104 bytes)
kerberos bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\kerberos.dll	876.50 KB (897,536 bytes)
msv1_0 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\msv1_0.dll	333.50 KB (341,504 bytes)

netlogon bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	936.00 KB (958,464 bytes)
w32time bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	540.50 KB (553,472 bytes)
iphlpapi bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	223.00 KB (228,352 bytes)
schannel bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	468.50 KB (479,744 bytes)
wdigest bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	161.50 KB (165,376 bytes)
rsaenh bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	371.83 KB (380,752 bytes)
rassfm bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	56.00 KB (57,344 bytes)
kdcsvc bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	571.50 KB (585,216 bytes)
ntdsa bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	3.82 MB (4,008,960 bytes)
ntdsatq bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	80.00 KB (81,920 bytes)
mswsock bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	671.50 KB (687,616 bytes)
esent bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	2.48 MB (2,604,544 bytes)
scecli bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	467.50 KB (478,720 bytes)
wshtcpip bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	38.00 KB (38,912 bytes)
ipsecsvc bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	410.50 KB (420,352 bytes)
oakley bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	493.50 KB (505,344 bytes)
winipsec bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	78.50 KB (80,384 bytes)

pstorsvc bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	56.00 KB (57,344 bytes)
psbase bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	162.50 KB (166,400 bytes)
dsenh bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	319.33 KB (326,992 bytes)
wlbsctrl bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	194.50 KB (199,168 bytes)
svchost bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	32.50 KB (33,280 bytes)
rpcss bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	645.00 KB (660,480 bytes)
clbcatq (1,292,800 bytes)	2001.12.4713.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	1.23 MB (1,292,800 bytes)
oleaut32	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	3.57 MB (3,739,136 bytes)
comres KB (798,208 bytes)	2001.12.4713.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	779.50 KB (798,208 bytes)
ntmarta bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	343.50 KB (351,744 bytes)
termsrv bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	607.00 KB (621,568 bytes)
icaapi bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	27.00 KB (27,648 bytes)
mstlsapi bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	311.00 KB (318,464 bytes)
activeds bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	543.50 KB (556,544 bytes)
adslrpc bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	312.00 KB (319,488 bytes)
credui bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	288.00 KB (294,912 bytes)
atl	3.00.2282 (348.00 KB (356,352 bytes)) 3/7/2003 9:10 AM	Microsoft Corporation	348.00 KB (356,352 bytes)
rdpwsx bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	256.63 KB (262,792 bytes)

schedsvc bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:42 AM	Microsoft Corporation	527.50 KB (540,160 bytes)
msidle bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	8.50 KB (8,704 bytes)
wkssvc bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	304.00 KB (311,296 bytes)
dmserver bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	45.50 KB (46,592 bytes)
srvc bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	188.00 KB (192,512 bytes)
sacsvr bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	27.50 KB (28,160 bytes)
trkwks bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	246.00 KB (251,904 bytes)
netman bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	591.00 KB (605,184 bytes)
mprapi bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	238.00 KB (243,712 bytes)
rtutils bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	81.50 KB (83,456 bytes)
rasapi32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	589.50 KB (603,648 bytes)
rasman bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	154.50 KB (158,208 bytes)
tapi32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	493.00 KB (504,832 bytes)
wzcsvc bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/6/2003 3:04 AM	Microsoft Corporation	608.50 KB (623,104 bytes)
wmi	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	5.00 KB (5,120 bytes)
dhcpcsvc bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	279.50 KB (286,208 bytes)
wzcsapi bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/6/2003 3:04 AM	Microsoft Corporation	49.50 KB (50,688 bytes)
netshell bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	2.65 MB (2,779,648 bytes)

clusapi bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	165.50 KB (169,472)
netcfgx bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	1.94 MB (2,034,688)
hnetcfg bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	764.00 KB (782,336)
wininet bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	1.43 MB (1,500,672)
wbemprox bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	46.00 KB (47,104)
wbemcomn bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	598.50 KB (612,864)
wmisvc bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	408.50 KB (418,304)
vssapi bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	1.28 MB (1,339,904)
wbemcore bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	1.64 MB (1,723,904)
esscli bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	919.50 KB (941,568)
fastprox bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	1.51 MB (1,580,544)
wbemsvc bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	62.50 KB (64,000)
es KB (652,800 bytes)	2001.12.4713.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	637.50
wmiutils bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	285.00 KB (291,840)
repdrvfs bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	595.50 KB (609,792)
comsvcs KB (3,106,816 bytes)	2001.12.4713.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	2.96 MB
wmiprvsd bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	1.38 MB (1,443,328)

sens bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	90.50 KB (92,672)
wbemess bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	976.50 KB (999,936)
ncprov bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	133.50 KB (136,704)
wbemcons bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	237.50 KB (243,200)
rasdlg bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	1.35 MB (1,420,800)
rasadhlp bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	13.00 KB (13,312)
pchsvc bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:42 AM	Microsoft Corporation	94.50 KB (96,768)
rdpsnd bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	63.00 KB (64,512)
scredir bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	67.50 KB (69,120)
cscui bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	574.00 KB (587,776)
drprov bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	26.50 KB (27,136)
msacm32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	77.00 KB (78,848)
msacm32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	252.00 KB (258,048)
ntlanman bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	108.00 KB (110,592)
netui0 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	181.50 KB (185,856)
netui1 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	482.00 KB (493,568)
davclnt bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	59.00 KB (60,416)
mprui bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	97.00 KB (99,328)

netui2 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	761.50 KB (779,776)
comdlg32 bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	706.00 KB (722,944)
netmsg bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	177.50 KB (181,760)
imaadp32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	55.00 KB (56,320)
msadp32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	49.00 KB (50,176)
msg711 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	33.00 KB (33,792)
msgsm32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	66.50 KB (68,096)
tssoft32	1.01 29.00 KB (29,696 bytes)	3/7/2003 9:11 AM	DSP GROUP, INC.
tsd32	c:\windows.2\system32\tssoft32.acm	Not Available 38.00 KB (38,912 bytes)	3/7/2003 9:11 AM
dfssvc bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	441.50 KB (452,096)
resutils bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	147.50 KB (151,040)
mfc42u	6.00.3014.0 3.34 MB (3,506,176 bytes)	3/7/2003 9:10 AM	Microsoft Corporation
wsock32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	23.00 KB (23,552)
rdpclip bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM	Microsoft Corporation	156.00 KB (159,744)
explorer bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	1.63 MB (1,704,960)
browseui bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	2.42 MB (2,536,960)
shdocvw bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM	Microsoft Corporation	3.20 MB (3,359,744)
apphelp bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM	Microsoft Corporation	262.50 KB (268,800)

themeui bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\themeui.dll	823.00 KB (842,752 bytes)
msimg32	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\msimg32.dll	7.00 KB (7,168 bytes)
linkinfo bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\linkinfo.dll	42.00 KB (43,008 bytes)
ntshrui bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\ntshrui.dll	233.50 KB (239,104 bytes)
webcheck bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\webcheck.dll	664.50 KB (680,448 bytes)
stobject bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\stobject.dll	178.00 KB (182,272 bytes)
batmeter bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\batmeter.dll	55.50 KB (56,832 bytes)
powrprof bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\powrprof.dll	36.00 KB (36,864 bytes)
urlmon bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\urlmon.dll	1.21 MB (1,271,296 bytes)
printui bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\printui.dll	1.09 MB (1,142,784 bytes)
cfgmgr32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\cfgmgr32.dll	16.00 KB (16,384 bytes)
cmd bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\cmd.exe	493.50 KB (505,344 bytes)
sqlopt	2000.080.0782.00 4/20/2003 8:26 AM Microsoft Corporation c:\program files\microsoft sql server\mssql\bin\sqlopt.exe	23.03 MB (24,153,600 bytes)
opens60	2000.080.0731.00 10/31/2002 1:34 AM Microsoft Corporation c:\program files\microsoft sql server\mssql\bin\opens60.dll	26.50 KB (27,136 bytes)
sqlsort	2000.080.0731.00 10/31/2002 1:35 AM Microsoft Corporation c:\program files\microsoft sql server\mssql\bin\sqlsort.dll	602.00 KB (616,448 bytes)
msvcirt bytes)	7.0.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\msvcirt.dll	161.50 KB (165,376 bytes)

sqllevn70	2000.080.0782.00 10/30/2002 2:44 PM Microsoft Corporation c:\program files\microsoft sql server\mssql\bin\sqllevn70.dll	19.50 KB (19,968 bytes)
xolehl	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM Microsoft Corporation c:\windows.2\system32\xolehl.dll	15.50 KB (15,872 bytes)
msdtcprx	2001.12.4713.0 (srv03_rtm.030305-2003) 3/7/2003 9:40 AM Microsoft Corporation c:\windows.2\system32\msdtcprx.dll	1.23 MB (1,285,120 bytes)
mtxcld	2001.12.4713.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\mtxcld.dll	199.00 KB (203,776 bytes)
winrnr	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\winrnr.dll	39.00 KB (39,936 bytes)
SSnetlib	2000.080.0782.00 4/15/2003 7:43 PM Microsoft Corporation c:\program files\microsoft sql server\mssql\bin\ssnetlib.dll	247.50 KB (253,440 bytes)
ssnmpn70	2000.080.0731.00 10/31/2002 1:35 AM Microsoft Corporation c:\program files\microsoft sql server\mssql\bin\ssnmpn70.dll	19.50 KB (19,968 bytes)
security	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\security.dll	7.00 KB (7,168 bytes)
SSmsLPCn	2000.080.0731.00 10/31/2002 1:35 AM Microsoft Corporation c:\program files\microsoft sql server\mssql\bin\ssmslpcn.dll	44.00 KB (45,056 bytes)
SSmsqlgc	2000.080.0782.00 4/15/2003 7:43 PM Microsoft Corporation c:\program files\microsoft sql server\mssql\bin\ssmsqlgc.dll	58.50 KB (59,904 bytes)
qlvipl	Not Available 3/18/2003 3:46 PM Not Available c:\windows.2\system32\qlvipl.dll	436.00 KB (446,464 bytes)
HelpCtr	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:42 AM Microsoft Corporation c:\windows.2\pchealth\helpctr\binaries\helpctr.exe	1.97 MB (2,066,432 bytes)
HCAAppRes	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:42 AM Microsoft Corporation c:\windows.2\pchealth\helpctr\binaries\hcappres.dll	6.00 KB (6,144 bytes)
itss	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\itss.dll	349.00 KB (357,376 bytes)
msxml3	8.40.9419.0 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\msxml3.dll	3.42 MB (3,590,656 bytes)
pchshell	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:42 AM Microsoft Corporation c:\windows.2\pchealth\helpctr\binaries\pchshell.dll	277.50 KB (284,160 bytes)

mlang bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\mlang.dll	799.00 KB (818,176 bytes)
mshtml	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\mshtml.dll	7.83 MB (8,208,384 bytes)
msimtf bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\msimtf.dll	528.00 KB (540,672 bytes)
msctf bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\msctf.dll	924.50 KB (946,688 bytes)
shdoclc bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\shdoclc.dll	588.00 KB (602,112 bytes)
jscrip	5.6.0.8515 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\jscrip.dll	1.21 MB (1,268,736 bytes)
msls31	3.10.349.0 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\msls31.dll	448.00 KB (458,752 bytes)
imm32 bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\imm32.dll	307.50 KB (314,880 bytes)
mshtmlm bytes)	6.00.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\mshtmlm.dll	1.34 MB (1,408,512 bytes)
vbscript	5.6.0.8515 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\vbscript.dll	1.06 MB (1,110,016 bytes)
mfc42	6.00.3014.0 3/7/2003 9:10 AM Microsoft Corporation c:\windows.2\system32\mfc42.dll	3.36 MB (3,526,656 bytes)
msinfo bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:42 AM Microsoft Corporation c:\windows.2\pchealth\helpctr\binaries\msinfo.dll	1.20 MB (1,257,984 bytes)
riched32	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\riched32.dll	5.00 KB (5,120 bytes)
riched20	5.31.23.1218 3/7/2003 9:11 AM Microsoft Corporation c:\windows.2\system32\riched20.dll	1.25 MB (1,313,280 bytes)
HelpSvc bytes)	5.2.3783.0 (srv03_rtm.030305-2003) 3/7/2003 9:42 AM Microsoft Corporation c:\windows.2\pchealth\helpctr\binaries\helpsvc.exe	2.18 MB (2,289,152 bytes)
[Services]		
Display Name	Name	State
Type	Path	Error Control
Alerter	Alerter	Running
		Auto
	c:\windows.2\system32\svchost.exe -k localservice	Share Process
	NT AUTHORITY\LocalService	Normal
		0

Application Layer Gateway Service	ALG	Stopped	Manual	
Own Process	c:\windows.2\system32\alg.exe		Normal	
NT AUTHORITY\LocalSystem	0			
Application Management Process	AppMgmt	Stopped	Manual	Share
c:\windows.2\system32\svchost.exe -k netsvcs			Normal	
LocalSystem	0			
Windows Audio	AudioSrv	Stopped	Disabled	Share
Process	c:\windows.2\system32\svchost.exe -k netsvcs		Normal	
LocalSystem	0			
Background Intelligent Transfer Service	BITS	Stopped	Manual	
Share Process	c:\windows.2\system32\svchost.exe			
-k netsvcs	Normal	LocalSystem	0	
Computer Browser	Browser	Stopped	Manual	Share
Process	c:\windows.2\system32\svchost.exe -k netsvcs		Normal	
LocalSystem	0			
Indexing Service	CISvc	Stopped	Disabled	Share
Process	c:\windows.2\system32\cisvc.exe		Normal	
LocalSystem	0			
ClipBook	ClipSrv	Stopped	Disabled	Own Process
c:\windows.2\system32\clipsrv.exe			Normal	
LocalSystem	0			
COM+ System Application	COMSysApp	Stopped	Manual	Own
Process	c:\windows.2\system32\dlhhost.exe /processid:{02d4b3f1-fd88-11d1-960d-00805fc79235}		Normal	LocalSystem
Cryptographic Services	CryptSvc	Stopped	Manual	Share
Process	c:\windows.2\system32\svchost.exe -k netsvcs		Normal	
LocalSystem	0			
Distributed File System	Dfs	Running	Auto	Own
Process	c:\windows.2\system32\dfsrv.exe		Normal	
LocalSystem	0			
DHCP Client	Dhcp	Running	Auto	Share Process
c:\windows.2\system32\svchost.exe -k networkservice				
Normal	NT AUTHORITY\NetworkService		0	
Logical Disk Manager Administrative Service			dmadmin	Stopped
Manual	Share Process			
c:\windows.2\system32\dmadmin.exe /com				Normal
LocalSystem	0			
Logical Disk Manager	dmserver	Running	Auto	Share
Process	c:\windows.2\system32\svchost.exe -k netsvcs		Normal	
LocalSystem	0			
DNS Client	Dnscache	Running	Auto	Share Process
c:\windows.2\system32\svchost.exe -k networkservice				
Normal	NT AUTHORITY\NetworkService		0	
Error Reporting Service	ERSvc	Stopped	Manual	Share
Process	c:\windows.2\system32\svchost.exe -k winerr		Ignore	
LocalSystem	0			
Event Log	Eventlog	Running	Auto	Share Process
c:\windows.2\system32\services.exe			Normal	
LocalSystem	0			

COM+ Event System	EventSystem	Running	Manual	Share
Process	c:\windows.2\system32\svchost.exe -k netsvcs		Normal	
Help and Support	System	Helpsvc	Running	Manual
Process	c:\windows.2\system32\svchost.exe -k netsvcs		Normal	
LocalSystem	0			
Human Interface Device Access		HidServ	Stopped	Disabled
Share Process		c:\windows.2\system32\svchost.exe		
-k netsvcs	Normal	LocalSystem	0	
HTTP SSL	HTTPFilter	Stopped	Manual	Share Process
c:\windows.2\system32\lsass.exe			Normal	
LocalSystem	0			
IAS Jet Database Access	IASJet	Stopped	Manual	Share
Process	c:\windows.2\syswow64\svchost.exe -k iasjet		Normal	
LocalSystem	0			
IMAPI CD-Burning COM Service		ImapiService	Stopped	Disabled
Own Process	c:\windows.2\system32\imapi.exe		Normal	
LocalSystem	0			
Intersite Messaging	IsmServ	Stopped	Disabled	Own
Process	c:\windows.2\system32\ismserv.exe		Normal	
LocalSystem	0			
Kerberos Key Distribution Center		kdc	Stopped	Disabled
Share Process		c:\windows.2\system32\lsass.exe		
Normal	LocalSystem	0		
Server	lanmanserver	Running	Auto	Share Process
c:\windows.2\system32\svchost.exe -k netsvcs			Normal	
LocalSystem	0			
Workstation	lanmanworkstation	Running	Auto	Share
Process	c:\windows.2\system32\svchost.exe -k netsvcs		Normal	
LocalSystem	0			
License Logging		LicenseService	Stopped	Disabled
Own Process	c:\windows.2\system32\lssrv.exe		Normal	
NT AUTHORITY\NetworkService		0		
TCP/IP NetBIOS Helper	LmHosts	Running	Auto	Share
Process	c:\windows.2\system32\svchost.exe -k localservice		Normal	
NT AUTHORITY\LocalService		0		
Messenger	Messenger	Stopped	Disabled	Share Process
c:\windows.2\system32\svchost.exe -k netsvcs			Normal	
LocalSystem	0			
Distributed Transaction Coordinator	MSDTC	Running	Auto	
Own Process	c:\windows.2\system32\msdtc.exe		Normal	
NT AUTHORITY\NetworkService		0		
Windows Installer	MSIServer	Stopped	Manual	Share
Process	c:\windows.2\system32\msiexec.exe /v		Normal	
LocalSystem	0			
MSSQL\$SETUPTEST	MSSQL\$SETUPTEST	Stopped	Manual	
Own Process	c:\program files\microsoft sql			
server\mssql\$setuptest\bin\sqlservr.exe -ssetuptest			Normal	
LocalSystem	0			
MSSQLSERVER	MSSQLSERVER	Stopped	Manual	
Own Process	c:\program files\microsoft sql			
server\mssql\bin\sqlservr.exe -smssqlserver			Normal	
LocalSystem	0			

MSSQLServerADHelper	MSSQLServerADHelper	Stopped	Manual	
Own Process	c:\program files\microsoft sql			
server\80\tools\bin\sqladhip.exe		Normal	LocalSystem	0
Network DDENetDDE	NetDDE	Stopped	Disabled	Share Process
c:\windows.2\system32\netdde.exe			Normal	
LocalSystem	0			
Network DDE DSDM	NetDDEdsdm	Stopped	Disabled	Share
Process	c:\windows.2\system32\netdde.exe		Normal	
LocalSystem	0			
Net Logon	Netlogon	Stopped	Manual	Share Process
c:\windows.2\system32\lsass.exe			Normal	
LocalSystem	0			
Network Connections	Netman	Running	Manual	Share
Process	c:\windows.2\system32\svchost.exe -k netsvcs		Normal	
LocalSystem	0			
Network Location Awareness (NLA)	Nla	Running	Manual	
Share Process		c:\windows.2\system32\svchost.exe		
-k netsvcs	Normal	LocalSystem	0	
File Replication	Ntfrs	Stopped	Manual	Own
Process	c:\windows.2\system32\ntfrs.exe		Ignore	
LocalSystem	0			
NT LM Security Support Provider	NtLmSsp	Stopped	Manual	
Share Process		c:\windows.2\system32\lsass.exe		
Normal	LocalSystem	0		
Removable Storage	NtmsSvc	Stopped	Manual	Share
Process	c:\windows.2\system32\svchost.exe -k netsvcs		Normal	
LocalSystem	0			
Plug and Play	PlugPlay	Running	Auto	Share Process
c:\windows.2\system32\services.exe			Normal	
LocalSystem	0			
IPSEC Services	PolicyAgent	Running	Auto	Share
Process	c:\windows.2\system32\lsass.exe		Normal	
LocalSystem	0			
Protected Storage	ProtectedStorage	Running	Auto	
Share Process		c:\windows.2\system32\lsass.exe		
Normal	LocalSystem	0		
Remote Access Auto Connection Manager			RasAuto	Stopped
Manual	Share Process			
c:\windows.2\system32\svchost.exe -k netsvcs				Normal
LocalSystem	0			
Remote Access Connection Manager	RasMan	Stopped	Manual	
Share Process		c:\windows.2\system32\svchost.exe		
-k netsvcs	Normal	LocalSystem	0	
Remote Desktop Help Session Manager	RDsessMgr	Stopped	Manual	
Own Process	c:\windows.2\system32\sessmgr.exe		Normal	
LocalSystem	0			
Routing and Remote Access			RemoteAccess	Stopped
Disabled	Share Process			
c:\windows.2\system32\svchost.exe -k netsvcs				Normal
LocalSystem	0			

Remote Registry	RemoteRegistry	Stopped	Manual
-k regsvc	Share Process	c:\windows.2\system32\svchost.exe	0
	Normal	NT AUTHORITY\LocalService	
Remote Procedure Call (RPC) Locator	RpcLocator	Stopped	Manual
	Own Process	c:\windows.2\system32\locator.exe	0
	NT AUTHORITY\NetworkService		Normal
Remote Procedure Call (RPC)	RpcSs	Running	Auto
rpcss	Share Process	c:\windows.2\system32\svchost.exe	-k
	Normal	LocalSystem	0
Resultant Set of Policy Provider	RSOPProv	Stopped	Manual
	Share Process	c:\windows.2\system32\rsopprov.exe	0
	LocalSystem		Normal
Special Administration Console Helper	sacsvr	Running	Manual
	Share Process	c:\windows.2\system32\svchost.exe	0
-k netsvcs	Normal	LocalSystem	0
Security Accounts ManagerSamSs	Running	Auto	Share
Process	c:\windows.2\system32\lsass.exe	0	Normal
	LocalSystem		
Smart Card	SCardSvr	Stopped	Manual
	c:\windows.2\system32\scardsvr.exe	0	Share Process
	LocalService		Ignore
	NT		
AUTHORITY\LocalService			
Task Scheduler	Schedule	Running	Auto
Process	c:\windows.2\system32\svchost.exe	-k	Share
	LocalSystem		Normal
	0		
Secondary Logon	seclogon	Stopped	Manual
Process	c:\windows.2\system32\svchost.exe	-k	Share
	LocalSystem		Ignore
	0		
System Event Notification	SENS	Running	Manual
Process	c:\windows.2\system32\svchost.exe	-k	Share
	LocalSystem		Normal
	0		
Shell Hardware Detection	ShellHWDetection	Stopped	Manual
	Share Process	c:\windows.2\system32\svchost.exe	0
-k netsvcs	Ignore	LocalSystem	0
Print Spooler	Spooler	Stopped	Manual
	c:\windows.2\system32\spoolsv.exe	0	Own Process
	LocalSystem		Normal
	0		
SQLAgent\$SETUPTEST	SQLAgent\$SETUPTEST	Stopped	Manual
	Own Process	c:\program files\microsoft sql server\mssql\$setup\test\bin\sqlagent.exe	-i
	LocalSystem		Normal
	0		
SQLSERVERAGENT	SQLSERVERAGENT	Stopped	Manual
	Own Process	c:\program files\microsoft sql server\mssql\bin\sqlagent.exe	-i
	LocalSystem		Normal
	0		
Windows Image Acquisition (WIA)	stisvc	Stopped	Disabled
	Share Process	c:\windows.2\system32\svchost.exe	0
-k imgsvc	Normal	NT AUTHORITY\LocalService	

Microsoft Software Shadow Copy Provider	swprv	Stopped
Performance Monitor	LocalSystem	0
Performance Log and Alert	SystemLog	Stopped
Own Process	c:\windows.2\system32\smlogsvc.exe	0
NT Authority\NetworkService		Normal
Telephony	TapiSrv	Stopped
	c:\windows.2\system32\svchost.exe	-k
	LocalSystem	0
Terminal Services	TermService	Running
Process	c:\windows.2\system32\svchost.exe	-k
	LocalSystem	0
Telnet	TlntSvr	Stopped
	c:\windows.2\system32\tlntsvr.exe	0
AUTHORITY\LocalService		Normal
Distributed Link Tracking Server	TrkSvr	Stopped
	Share Process	c:\windows.2\system32\svchost.exe
-k netsvcs	Normal	LocalSystem
Distributed Link Tracking Client	TrkWks	Running
	Share Process	c:\windows.2\system32\svchost.exe
-k netsvcs	Normal	LocalSystem
Terminal Services Session Directory	Tssdis	Stopped
	Own Process	c:\windows.2\system32\tssdis.exe
	LocalSystem	0
Upload Manager	uploadmgr	Stopped
Process	c:\windows.2\system32\svchost.exe	-k
	LocalSystem	0
Uninterruptible Power Supply	UPS	Stopped
	Own Process	c:\windows.2\system32\ups.exe
	NT AUTHORITY\LocalService	0
Virtual Disk Service	vds	Stopped
Process	c:\windows.2\system32\vds.exe	0
	LocalSystem	0
Volume Shadow Copy	VSS	Stopped
Process	c:\windows.2\system32\vssvc.exe	0
	LocalSystem	0
Windows Time	W32Time	Running
Process	c:\windows.2\system32\svchost.exe	-k
	LocalSystem	0
WebClient	WebClient	Stopped
	c:\windows.2\system32\svchost.exe	-k
	NT AUTHORITY\LocalService	0
WinHTTP Web Proxy Auto-Discovery Service	WinHttpAutoProxySvc	Stopped
	Manual	Share Process
	c:\windows.2\system32\svchost.exe	-k
	NT AUTHORITY\LocalService	0
Windows Management Instrumentation	winmgmt	Running
	Share Process	c:\windows.2\system32\svchost.exe
-k netsvcs	Ignore	LocalSystem
Windows Management Instrumentation Driver Extensions		Wmi
	Stopped	Manual
	c:\windows.2\system32\svchost.exe	-k
	LocalSystem	0

WMI Performance Adapter	WmiApSrv	Stopped	Manual	Own
Process	c:\windows.2\system32\wbem\wmiaprv.exe	0	Normal	
Automatic Updates	WuauServ	Stopped	Manual	Share
Process	c:\windows.2\system32\svchost.exe	-k	Normal	
	LocalSystem	0		
Wireless Configuration	WZCSVC	Stopped	Disabled	Share
Process	c:\windows.2\system32\svchost.exe	-k	Normal	
	LocalSystem	0		

[Program Groups]

Group Name	Name	User Name
Accessories	Default User:Accessories	Default User
Accessories\Entertainment	Default User:Accessories\Entertainment	Default User
Accessories\Accessibility	Default User:Accessories\Accessibility	Default User
Startup	Default User:Startup	Default User
Startup	All Users:Startup	All Users
Accessories	All Users:Accessories	All Users
Accessories\Communications	All Users:Accessories\Communications	All Users
Accessories\System Tools	All Users:Accessories\System Tools	All Users
Accessories\Entertainment	All Users:Accessories\Entertainment	All Users
Accessories\Accessibility	All Users:Accessories\Accessibility	All Users
Administrative Tools	All Users:Administrative Tools	All Users
Microsoft SQL Server	All Users:Microsoft SQL Server	All Users

Accessories	NT AUTHORITY\SYSTEM:Accessories	NT AUTHORITY\SYSTEM
Accessories\Entertainment	NT AUTHORITY\SYSTEM:Accessories\Entertainment	NT AUTHORITY\SYSTEM
Accessories\Accessibility	NT AUTHORITY\SYSTEM:Accessories\Accessibility	NT AUTHORITY\SYSTEM
Startup	NT AUTHORITY\SYSTEM:Startup	NT AUTHORITY\SYSTEM
Accessories	SQLORCA\Administrator:Accessories	SQLORCA\Administrator
Accessories\Entertainment	SQLORCA\Administrator:Accessories\Entertainment	SQLORCA\Administrator
Accessories\Accessibility	SQLORCA\Administrator:Accessories\Accessibility	SQLORCA\Administrator
Startup	SQLORCA\Administrator:Startup	SQLORCA\Administrator
Administrative Tools	SQLORCA\Administrator:Administrative Tools	SQLORCA\Administrator

QLogic Corporation SQLORCA\Administrator:QLogic Corporation
 SQLORCA\Administrator
 QLogic Corporation\SANblade Control VIX
 SQLORCA\Administrator:QLogic Corporation\SANblade Control
 VIX SQLORCA\Administrator

[Startup Programs]

Program	Command	User Name	Location
desktop	desktop.ini	NT AUTHORITY\SYSTEM	Startup
desktop	desktop.ini	SQLORCA\Administrator	Startup
desktop	desktop.ini	.DEFAULT	Startup
desktop	desktop.ini	All Users	Common Startup

[OLE Registration]

Object	Local Server
Sound (OLE2)	sndrec32.exe
Media Clip	mplay32.exe
Video Clip	mplay32.exe /avi
MIDI Sequence	mplay32.exe /mid
Sound	Not Available
Media Clip	Not Available
WordPad Document	"%programfiles%\windows nt\accessories\wordpad.exe"
Bitmap Image	mspaint.exe

[Windows Error Reporting]

Time	Type	Details
------	------	---------

[Internet Settings]

[Internet Explorer]

[Following are sub-categories of this main category]

[Summary]

Item	Value
Version	6.0.3783.0
Build	63783
Application Path	C:\Program Files\Internet Explorer
Language	English (United States)
Active Printer	Not Available

Cipher Strength	128-bit
Content Advisor	Disabled
IEAK Install	No

[File Versions]

File	Version	Size	Date	Path	Company
actxprxy.dll	6.0.3783.0	230 KB	3/6/2003 3:02:12 AM	C:\WINDOWS.2\system32	Microsoft Corporation
advpack.dll	6.0.3783.0	240 KB	3/6/2003 3:02:14 AM	C:\WINDOWS.2\system32	Microsoft Corporation
advpack.dll	6.0.3787.0	240 KB	3/13/2003 1:36:56 AM	C:\jdw	Microsoft Corporation
asctrls.ocx	6.0.3783.0	219 KB	3/6/2003 3:01:16 AM	C:\WINDOWS.2\system32	Microsoft Corporation
browsecl.dll	6.0.3783.0	62 KB	3/6/2003 2:58:00 AM	C:\WINDOWS.2\system32	Microsoft Corporation
browseui.dll	6.0.3783.0	2,478 KB	3/6/2003 3:02:22 AM	C:\WINDOWS.2\system32	Microsoft Corporation
cdfview.dll	6.0.3783.0	292 KB	3/6/2003 3:02:22 AM	C:\WINDOWS.2\system32	Microsoft Corporation
comctl32.dll	5.82.3783.0	1,584 KB	3/6/2003 3:02:32 AM	C:\WINDOWS.2\system32	Microsoft Corporation
dxtrans.dll	6.3.3783.0	562 KB	3/6/2003 3:02:48 AM	C:\WINDOWS.2\system32	Microsoft Corporation
dxtmsft.dll	6.3.3783.0	918 KB	3/6/2003 3:02:48 AM	C:\WINDOWS.2\system32	Microsoft Corporation
iecont.dll	<File Missing>		Not Available	Not Available	Not Available
iecontlc.dll	<File Missing>		Not Available	Not Available	Not Available
iedkcs32.dll	16.0.3783.0	676 KB	3/6/2003 3:03:04 AM	C:\WINDOWS.2\system32	Microsoft Corporation
iepeers.dll	6.0.3783.0	652 KB	3/6/2003 3:03:04 AM	C:\WINDOWS.2\system32	Microsoft Corporation
iesetup.dll	6.0.3783.0	89 KB	3/6/2003 3:03:04 AM	C:\WINDOWS.2\system32	Microsoft Corporation
ieuinit.inf	Not Available	20 KB	3/5/2003 10:08:54 PM	C:\WINDOWS.2\system32	Not Available
ieexplore.exe	6.0.3783.0	102 KB	3/6/2003 3:01:40 AM	C:\Program Files\Internet Explorer	Microsoft Corporation
imgutil.dll	5.2.3783.0	101 KB	3/6/2003 3:03:06 AM	C:\WINDOWS.2\system32	Microsoft Corporation
inetcpl.cpl	6.0.3783.0	589 KB	3/6/2003 3:01:12 AM	C:\WINDOWS.2\system32	Microsoft Corporation
inetcpl.dll	6.0.3783.0	108 KB	3/6/2003 2:59:20 AM	C:\WINDOWS.2\system32	Microsoft Corporation
inseng.dll	6.0.3783.0	213 KB	3/6/2003 3:03:08 AM	C:\WINDOWS.2\system32	Microsoft Corporation
mlang.dll	6.0.3783.0	799 KB	3/6/2003 3:03:24 AM	C:\WINDOWS.2\system32	Microsoft Corporation
msencode.dll	<File Missing>		Not Available	Not Available	Not Available
mshsta.exe	6.0.3783.0	59 KB	3/6/2003 3:01:48 AM	C:\WINDOWS.2\system32	Microsoft Corporation
mshtml.dll	6.0.3783.0	8,016 KB	3/6/2003 3:03:38 AM	C:\WINDOWS.2\system32	Microsoft Corporation

mshtml.tlb	6.0.3783.0	1,319 KB	3/5/2003 9:35:20 PM	C:\WINDOWS.2\system32	Microsoft Corporation
mshtmlmed.dll	6.0.3783.0	235 KB	3/6/2003 3:03:38 AM	C:\WINDOWS.2\system32	Microsoft Corporation
mshtmlr.dll	6.0.3783.0	56 KB	3/6/2003 2:59:46 AM	C:\WINDOWS.2\system32	Microsoft Corporation
msident.dll	6.0.3783.0	128 KB	3/6/2003 3:03:38 AM	C:\WINDOWS.2\system32	Microsoft Corporation
msidntld.dll	6.0.3783.0	14 KB	3/6/2003 2:59:46 AM	C:\WINDOWS.2\system32	Microsoft Corporation
msieftp.dll	6.0.3783.0	536 KB	3/6/2003 3:03:38 AM	C:\WINDOWS.2\system32	Microsoft Corporation
msrating.dll	6.0.3783.0	379 KB	3/6/2003 3:03:42 AM	C:\WINDOWS.2\system32	Microsoft Corporation
mstime.dll	6.0.3783.0	1,621 KB	3/6/2003 3:03:42 AM	C:\WINDOWS.2\system32	Microsoft Corporation
occache.dll	6.0.3783.0	201 KB	3/6/2003 3:03:56 AM	C:\WINDOWS.2\system32	Microsoft Corporation
proctexe.ocx	<File Missing>		Not Available	Not Available	Not Available
sendmail.dll	6.0.3783.0	97 KB	3/6/2003 3:04:22 AM	C:\WINDOWS.2\system32	Microsoft Corporation
shdoclc.dll	6.0.3783.0	588 KB	3/6/2003 2:57:28 AM	C:\WINDOWS.2\system32	Microsoft Corporation
shdocvw.dll	6.0.3783.0	3,281 KB	3/6/2003 3:04:22 AM	C:\WINDOWS.2\system32	Microsoft Corporation
shfolder.dll	6.0.3783.0	37 KB	3/6/2003 3:04:22 AM	C:\WINDOWS.2\system32	Microsoft Corporation
shlwapi.dll	6.0.3783.0	722 KB	3/6/2003 3:04:24 AM	C:\WINDOWS.2\system32	Microsoft Corporation
tdc.ocx	1.3.0.3130	177 KB	3/6/2003 3:01:18 AM	C:\WINDOWS.2\system32	Microsoft Corporation
url.dll	6.0.3783.0	45 KB	3/6/2003 3:04:36 AM	C:\WINDOWS.2\system32	Microsoft Corporation
urlmon.dll	6.0.3783.0	1,242 KB	3/6/2003 3:04:38 AM	C:\WINDOWS.2\system32	Microsoft Corporation
webcheck.dll	6.0.3783.0	665 KB	3/6/2003 3:04:44 AM	C:\WINDOWS.2\system32	Microsoft Corporation
wininet.dll	6.0.3783.0	1,466 KB	3/6/2003 3:04:46 AM	C:\WINDOWS.2\system32	Microsoft Corporation

[Connectivity]

Item	Value
Connection Preference	Never dial

LAN Settings

AutoConfigProxy	Not Available
AutoProxyDetectMode	Disabled
AutoConfigURL	
Proxy	Enabled
ProxyServer	http://itgproxy:80
ProxyOverride	<local>

[Cache]

[Following are sub-categories of this main category]
[Summary]

Item	Value
Page Refresh Type	Automatic
Temporary Internet Files Folder	C:\Documents and Settings\NetworkService.NT AUTHORITY.001\Local Settings\Temporary Internet Files
Total Disk Space	Not Available
Available Disk Space	Not Available
Maximum Cache Size	Not Available
Available Cache Size	Not Available

[List of Objects]

Program File Status CodeBase
No cached object information available

[Content]

[Following are sub-categories of this main category]
[Summary]

Item	Value
Content Advisor	Disabled

[Personal Certificates]

Issued To Issued By Validity Signature Algorithm
No personal certificate information available

[Other People Certificates]

Issued To Issued By Validity Signature Algorithm
No other people certificate information available

[Publishers]

Name
No publisher information available

[Security]

Zone Security Level
My Computer Custom
Local intranetMedium-low
Trusted sites Low

Restricted sites Medium High

Microsoft SQL Server 8.0 Startup Parameters

Set LOGAFFINITY = 0
sqlservr -c -x -T888 -T825 -T827 -T828 -T3502 -T3428

where
T825: enable NUMA support
T826: don't use large pages for buffer array
T827: enable Super Latches
T828: don't do timed checkpoints
T888: use locked pages for buffers

Environment variable LOGAFFINITY: specifies which processor the log writer will be affinityized to.

U12ser Rights Assignment

The Group Policy Editor of Windows.net was used to modify an entry under User Rights Assignment. Specifically, the right to "Lock pages in memory" was given to the Administrators group so that SQL Server 2000 could use large amounts of physical memory.

C.1 Microsoft SQL Server 8.0 Configuration Parameters

name	minimum	maximum	config_value	run_value
affinity mask	-2147483648	2147483647	-2	-2
affinity64 mask	-2147483648	2147483647	-1	-1
allow updates	0	1	0	0
awe enabled	0	1	0	0
c2 audit mode	0	1	0	0
cost threshold for parallelism	0	32767	5	5
Cross DB Ownership Chaining	0	1	0	0
cursor threshold	-1	2147483647	-1	-1
default full-text language	0	2147483647	1033	1033
default language	0	9999	0	0
fill factor (%)	0	100	0	0

index create memory (KB)	704	2147483647	0	0
lightweight pooling	0	1	1	1
locks	5000	2147483647	0	0
max degree of parallelism	0	32	1	1
max server memory (MB)	4	2147483647	504000	
max text repl size (B)	0	2147483647	65536	65536
max worker threads	32	32767	1450	1450
media retention	0	365	0	0
min memory per query (KB)	512	2147483647	1024	
min server memory (MB)	0	2147483647	0	0
nested triggers	0	1	1	1
network packet size (B)	512	65536	4096	4096
open objects	0	2147483647	0	0
priority boost	0	1	1	1
query governor cost limit	0	2147483647	0	0
query wait (s)	-1	2147483647	-1	-1
recovery interval (min)	0	32767	119	119
remote access	0	1	1	1
remote login timeout (s)	0	2147483647	20	20
remote proc trans	0	1	0	0
remote query timeout (s)	0	2147483647	600	600
scan for startup procs	0	1	0	0
set working set size	0	1	0	0
show advanced options	0	1	1	1
two digit year cutoff	1753	9999	2049	2049
user connections	0	32767	0	0
user options	0	32767	0	0

C.2 Client System Configuration Parameters

COM+ Settings

TPCC.AllTxns:
Activation:
 Enable Object Pooling selected
 Minimum Pool Size: 42
 Maximum Pool Size: 42
 Creation Timeout: 60000
 Enable Object Construction
 Enable Just In Time Activation
Concurrency:
 Concurrency Required

Microsoft IIS Registry Parameters

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00021500
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,00,00
"PoolThreadLimit"=dword:0002000
"ThreadTimeout"=dword:00086400
"MaxConnections"=dword:00020000

World Wide Web Service Registry Parameters

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters
Class Name: <NO CLASS>
Last Write Time: 5/10/2002 - 2:37 PM

Value 0
Name: AcceptExOutstanding
Type: REG_DWORD
Data: 0x28

Value 1
Name: AccessDeniedMessage
Type: REG_SZ
Data: Error: Access is Denied.

Value 2
Name: CertMapList
Type: REG_SZ
Data: C:\WINNT\System32\inetrv\iisrmap.dll

Value 3
Name: Filter DLLs
Type: REG_SZ
Data:

Value 4
Name: InstallPath
Type: REG_SZ
Data: C:\WINNT\System32\inetrv

Value 5
Name: LogFileDirectory
Type: REG_SZ
Data: C:\WINNT\System32\LogFiles

Value 6
Name: MajorVersion
Type: REG_DWORD
Data: 0x5

Value 7
Name: MinorVersion
Type: REG_DWORD
Data: 0

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch
Class Name: <NO CLASS>
Last Write Time: 5/6/2002 - 9:05 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory
Class Name: <NO CLASS>
Last Write Time: 5/6/2002 - 9:05 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSERVER.DataFactory
Class Name: <NO CLASS>
Last Write Time: 5/6/2002 - 9:05 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map
Class Name: <NO CLASS>
Last Write Time: 5/7/2002 - 9:42 AM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots
Class Name: <NO CLASS>
Last Write Time: 6/28/2002 - 5:33 PM

Value 0
Name: /
Type: REG_SZ
Data: c:\inetpub\wwwroot,,205

Value 1
Name: /_vti_bin
Type: REG_SZ
Data: C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\40\isapi,,1

Value 2
Name: /IISAdmin
Type: REG_SZ
Data: C:\WINNT\System32\inetrv\iisadmin,,1

Value 3
Name: /IISHelp
Type: REG_SZ
Data: c:\winnt\help\iishelp,,1

Value 4
Name: /IISSamples
Type: REG_SZ
Data: c:\inetpub\iissamples,,1

Value 5
Name: /MSADC
Type: REG_SZ
Data: c:\program files\common files\system\msadc,,1

Value 6
Name: /Printers
Type: REG_SZ
Data: C:\WINNT\web\printers,,201

Value 7
Name: /Scripts
Type: REG_SZ
Data: c:\inetpub\scripts,,1

TPCC Application Registry Parameters

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPCC]
"Path"="C:\inetpub\wwwroot\
"NumberOfDeliveryThreads"=dword:00000012
"MaxConnections"=dword:00021500
"MaxPendingDeliveries"=dword:00002000
"DB_Protocol"="ODBC"
"TxnMonitor"="COM"
"DbServer"="sqlorca"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"="*****"
"COM_SinglePool"="YES"
"SPPrefix"="SP0"

TCPIP Registry Parameters

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters]
"NV Hostname"="tpcweb1"
"DataBasePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,\
00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,\
64,00,72,00,69,00,76,00,65,00,72,00,73,00,5c,00,65,00,74,00,63,00,00,00
"NameServer"=""
"ForwardBroadcasts"=dword:00000000

"EnableRouter"=dword:00000000
"Hostname"="tpcweb1"
"SearchList"=""
"UseDomainNameDevolution"=dword:00000001
"EnableICMPRedirect"=dword:00000001
"DeadGWDetectDefault"=dword:00000001
"DontAddDefaultGatewayDefault"=dword:00000000
"EnableSecurityFilters"=dword:00000000
"AllowUnqualifiedQuery"=dword:00000000
"PrioritizeRecordData"=dword:00000001
"MaxUserPort"=dword:0000fffe
"DhcpNameServer"="15.1.100.1"
"DhcpDomain"="hp-perf.net"

Microsoft Windows 2000 Server Configuration Parameters

System Information report written at: 03/04/2003 06:41:11 PM
[System Information]

[Following are sub-categories of this main category]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Server
Version	5.0.2195 Service Pack 3 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	W11
System Manufacturer	HP
System Model	ProLiant DL360 G3
System Type	X86-based PC
Processor	x86 Family 15 Model 2 Stepping 7 GenuineIntel ~2799 Mhz
Processor	x86 Family 15 Model 2 Stepping 7 GenuineIntel ~2799 Mhz
Processor	x86 Family 15 Model 2 Stepping 7 GenuineIntel ~2799 Mhz
Processor	x86 Family 15 Model 2 Stepping 7 GenuineIntel ~2799 Mhz
BIOS Version	02/07/03
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition1
Locale	United States
User Name	W11\Administrator
Time Zone	Pacific Standard Time
Total Physical Memory	1,048,084 KB
Available Physical Memory	876,808 KB
Total Virtual Memory	4,093,552 KB

Available Space Memory 3,845,758 KB
Page File C:\pagefile.sys

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

Resource Device
No conflicted/shared resources

[DMA]

Channel	Device	Status
7	Direct memory access controller	OK
2	Standard floppy disk controller	OK

[Forced Hardware]

Device PNP Device ID
No Forced Hardware

[I/O]

Address Range	Device	Status
0x0000-0x0CFF	PCI bus	OK
0x0000-0x0CFF	PCI bus	OK
0x0000-0x0CFF	Direct memory access controller	OK
0x03B0-0x03BB	PCI bus	OK
0x03B0-0x03BB	ATI Technologies Inc. RAGE XL PCI	OK
0x03C0-0x03DF	PCI bus	OK
0x03C0-0x03DF	ATI Technologies Inc. RAGE XL PCI	OK
0x2400-0x24FF	ATI Technologies Inc. RAGE XL PCI	OK
0x2800-0x28FF	Compaq Smart Array 5i	OK
0x1800-0x18FF	hp ProLiant iLO Advanced System Management Controller	OK
0x2C00-0x2CFF	HP iLO Management Interface Driver	OK
0x0A79-0x0A79	ISAPNP Read Data Port	OK
0x0279-0x0279	ISAPNP Read Data Port	OK
0x02F4-0x02F7	ISAPNP Read Data Port	OK
0x0F50-0x0F58	Motherboard resources	OK
0x0408-0x040F	Motherboard resources	OK
0x0092-0x0092	Motherboard resources	OK
0x0900-0x0903	Motherboard resources	OK
0x0910-0x0911	Motherboard resources	OK
0x0920-0x0923	Motherboard resources	OK
0x0930-0x0937	Motherboard resources	OK
0x0940-0x0947	Motherboard resources	OK
0x0950-0x0957	Motherboard resources	OK
0x0C06-0x0C08	Motherboard resources	OK
0x0C14-0x0C14	Motherboard resources	OK
0x0C49-0x0C4A	Motherboard resources	OK

0x0C50-0x0C52	Motherboard resources	OK
0x0C6C-0x0C6F	Motherboard resources	OK
0x0010-0x001F	Motherboard resources	OK
0x0230-0x0233	Motherboard resources	OK
0x0260-0x0267	Motherboard resources	OK
0x04D0-0x04D1	Motherboard resources	OK
0x0700-0x070F	Motherboard resources	OK
0x0800-0x081F	Motherboard resources	OK
0x0C80-0x0C83	Motherboard resources	OK
0x0CD4-0x0CD7	Motherboard resources	OK
0x0CF9-0x0CF9	Motherboard resources	OK
0x0020-0x0021	Programmable interrupt controller	OK
0x00A0-0x00A1	Programmable interrupt controller	OK
0x0C00-0x0C01	Programmable interrupt controller	OK
0x0040-0x0043	System timer	OK
0x0080-0x008F	Direct memory access controller	OK
0x00C0-0x00DF	Direct memory access controller	OK
0x040B-0x040B	Direct memory access controller	OK
0x04D6-0x04D6	Direct memory access controller	OK
0x0061-0x0061	System speaker	OK
0x0060-0x0060	Standard 101/102-Key or Microsoft Natural PS/2	
Keyboard OK		
0x0064-0x0064	Standard 101/102-Key or Microsoft Natural PS/2	
Keyboard OK		
0x002E-0x002F	Extended IO Bus	OK
0x0220-0x0223	Extended IO Bus	OK
0x0240-0x025F	Extended IO Bus	OK
0x0070-0x0073	Extended IO Bus	OK
0x03F8-0x03FF	Communications Port (COM1)	OK
0x03F2-0x03F5	Standard floppy disk controller	OK
0x03F7-0x03F7	Standard floppy disk controller	OK
0x2000-0x200F	Standard Dual Channel PCI IDE Controller	
OK		
0x01F0-0x01F7	Primary IDE Channel	OK
0x03F6-0x03F6	Primary IDE Channel	OK
0x0170-0x0177	Secondary IDE Channel	OK
0x0376-0x0376	Secondary IDE Channel	OK
0x3000-0x30FF	PCI bus	OK
0x3000-0x30FF	QLogic QLA23xx PCI Fibre Channel Adapter	
OK		

[IRQs]

IRQ Number	Device
9	Microsoft ACPI-Compliant System
31	Compaq Smart Array 5i
23	hp ProLiant iLO Advanced System Management Controller
22	HP iLO Management Interface Driver
1	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
12	PS/2 Compatible Mouse
4	Communications Port (COM1)
6	Standard floppy disk controller

74	Standard OpenHCD USB Host Controller
28	QLogic QLA23xx PCI Fibre Channel Adapter
30	Compaq NC7781 Gigabit Server Adapter #2
29	Compaq NC7781 Gigabit Server Adapter

[Memory]

Range	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	ATI Technologies Inc. RAGE XL PCI	OK
0xF5D00000-0xF6FFFFFF	PCI bus	OK
0xF6000000-0xF6FFFFFF	ATI Technologies Inc. RAGE XL PCI	OK
0xF5FF0000-0xF5FF0FFF	ATI Technologies Inc. RAGE XL PCI	OK
0xF5F80000-0xF5FBFFFF	Compaq Smart Array 5i	OK
0xF5DF0000-0xF5DF3FFF	Compaq Smart Array 5i	OK
0xF5F70000-0xF5F701FF	hp ProLiant iLO Advanced System Management Controller	OK
0xF5F60000-0xF5F607FF	HP iLO Management Interface Driver	OK
0xF5F50000-0xF5F51FFF	HP iLO Management Interface Driver	OK
0xF5E80000-0xF5E8FFFF	HP iLO Management Interface Driver	OK
0xF5E70000-0xF5E70FFF	Standard OpenHCD USB Host Controller	OK
0xF7E00000-0xF7EFFFFFF	PCI bus	OK
0xF7EF0000-0xF7EF0FFF	QLogic QLA23xx PCI Fibre Channel Adapter	
OK		
0xF7EE0000-0xF7EEFFFF	Compaq NC7781 Gigabit Server Adapter #2	
OK		
0xF7F00000-0xF7FFFFFF	PCI bus	OK
0xF7FF0000-0xF7FFFFFF	Compaq NC7781 Gigabit Server Adapter	OK

[Components]

[Following are sub-categories of this main category]

[Multimedia]

[Following are sub-categories of this main category]

[Audio Codecs]

Codec	Manufacturer	Description	Status	File	Version
		Size	Creation Date		
c:\winnt\system32\msadp32.acm	Microsoft Corporation				
OK	C:\WINNT\System32\MSADP32.ACM				
5.00.2134.1	14.77 KB (15,120 bytes)	12/31/1979 4:00:00			
PM					
c:\winnt\system32\lhacm.acm	Microsoft Corporation				
OK	C:\WINNT\System32\LHACM.ACM				4.4.3385
33.27 KB (34,064 bytes)	4/2/2003 11:35:54 AM				
c:\winnt\system32\msgsm32.acm	Microsoft Corporation				
OK	C:\WINNT\System32\MSGSM32.ACM				
5.00.2134.1	22.27 KB (22,800 bytes)	12/31/1979 4:00:00			
PM					

c:\winnt\system32\msg711.acm	Microsoft Corporation				
OK	C:\WINNT\System32\MSG711.ACM				
5.00.2134.1	10.27 KB (10,512 bytes)	12/31/1979 4:00:00			
PM					
c:\winnt\system32\tssoft32.acm	DSP GROUP, INC.				
OK	C:\WINNT\System32\TSSOFT32.ACM				1.01
9.27 KB (9,488 bytes)	12/31/1979 4:00:00 PM				
c:\winnt\system32\msg723.acm	Microsoft Corporation				
OK	C:\WINNT\System32\MSG723.ACM				4.4.3385
106.77 KB (109,328 bytes)	4/2/2003 11:35:54 AM				
c:\winnt\system32\iac25_32.ax	Intel Corporation				Indeo®
audio software	OK	C:\WINNT\System32\IAC25_32.AX			
2.05.53	195.00 KB (199,680 bytes)	12/31/1979 4:00:00			
PM					
c:\winnt\system32\imaadp32.acm	Microsoft Corporation				
OK	C:\WINNT\System32\IMAADP32.ACM				
5.00.2134.1	16.27 KB (16,656 bytes)	12/31/1979 4:00:00			
PM					
[Video Codecs]					
Codec	Manufacturer	Description	Status	File	Version
		Size	Creation Date		
c:\winnt\system32\ir50_32.dll	Intel Corporation				Indeo®
video 5.10	OK	C:\WINNT\System32\IR50_32.DLL			
R.5.10.15.2.55	737.50 KB (755,200 bytes)	12/31/1979 4:00:00 PM			
c:\winnt\system32\msh263.drv	Microsoft Corporation				
OK	C:\WINNT\System32\MSH263.DRV				4.4.3385
252.27 KB (258,320 bytes)	4/2/2003 11:35:46 AM				
c:\winnt\system32\msrle32.dll	Microsoft Corporation				
OK	C:\WINNT\System32\MSRLE32.DLL				
5.00.2134.1	10.77 KB (11,024 bytes)	12/31/1979 4:00:00			
PM					
c:\winnt\system32\msvidc32.dll	Microsoft Corporation				
OK	C:\WINNT\System32\MSVIDC32.DLL				
5.00.2134.1	27.27 KB (27,920 bytes)	12/31/1979 4:00:00			
PM					
c:\winnt\system32\msh261.drv	Microsoft Corporation				
OK	C:\WINNT\System32\MSH261.DRV				4.4.3385
163.77 KB (167,696 bytes)	4/2/2003 11:35:54 AM				
c:\winnt\system32\ir32_32.dll	Intel(R) Corporation				
OK	C:\WINNT\System32\IR32_32.DLL				Not
Available	194.50 KB (199,168 bytes)	12/31/1979 4:00:00 PM			
c:\winnt\system32\iccvld.dll	Radius Inc.				OK
C:\WINNT\System32\ICCVLD.DLL	1.10.0.6	108.00			
KB (110,592 bytes)	12/31/1979 4:00:00 PM				
[CD-ROM]					
Item	Value				
Drive	D:				

Description CD-ROM Drive
Media Loaded False
Media Type CD-ROM
Name COMPAQ CRN-8245B
Manufacturer (Standard CD-ROM drives)
Status OK
Transfer RateNot Available
SCSI Target ID 0
PNP Device ID IDE\CDROMCOMPAQ_CRN-8245B_2.19\5\FB0C83D&0&0.0.0

[Sound Device]

Item Value
No sound devices

[Display]

Item Value
Name ATI Technologies Inc. RAGE XL PCI
PNP Device ID PCI\VEN_1002&DEV_4752&SUBSYS_001E0E11&REV_27\3&267A616A&0&18
Adapter TypeATI RAGE XL PCI, ATI Technologies Inc. compatible
Adapter Description ATI Technologies Inc. RAGE XL PCI
Adapter RAM 8.00 MB (8,388,608 bytes)
Installed Drivers atidrab.dll
Driver Version 5.00.2179.1
INF File display.inf (atirage3 section)
Color Planes 1
Color Table Entries 65536
Resolution 640 x 480 x 60 hertz
Bits/Pixel 16

[Infrared]

Item Value
No infrared devices

[Input]

[Following are sub-categories of this main category]

[Keyboard]

Item Value
Description Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
Name Enhanced (101- or 102-key)
Layout 00000409
PNP Device ID ACPI\PNP0303\4&35118DFF&0

NumberOfFunctionKeys 12

[Pointing Device]

Item Value
Hardware Type PS/2 Compatible Mouse
Number of Buttons 2
Status OK
PNP Device ID ACPI\PNP0F13\4&35118DFF&0
Power Management Supported False
Double Click Threshold 6
Handedness Right Handed Operation

[Modem]

Item Value
No modems

[Network]

[Following are sub-categories of this main category]

[Adapter]

Item Value
Name [00000000] Compaq NC7781 Gigabit Server Adapter
Adapter TypeEthernet 802.3
Product Name Compaq NC7781 Gigabit Server Adapter
Installed True
PNP Device ID PCI\VEN_14E4&DEV_16A7&SUBSYS_00CB0E11&REV_02\3&1070020&0&10
Last Reset 3/3/2003 5:56:57 PM
Index 0
Service Name q57w2k
IP Address 15.1.100.112
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled True
DHCP Server 15.1.100.1
DHCP Lease Expires 3/12/2003 1:57:41 AM
DHCP Lease Obtained 3/4/2003 1:57:41 AM
MAC Address 00:0B:CD:50:42:39
Service Name q57w2k
IRQ Number 29
Driver c:\winnt\system32\drivers\q57w2k.sys (77776, 2.75.0.0)

Name [00000001] Compaq NC7781 Gigabit Server Adapter
Adapter TypeEthernet 802.3
Product Name Compaq NC7781 Gigabit Server Adapter
Installed True

PNP Device ID PCI\VEN_14E4&DEV_16A7&SUBSYS_00CB0E11&REV_02\3&13C0B0C5&0&10
Last Reset 3/3/2003 5:56:57 PM
Index 1
Service Name q57w2k
IP Address
IP Subnet
Default IP Gateway Not Available
DHCP Enabled True
DHCP Server
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:0B:CD:50:41:F1
Service Name q57w2k
IRQ Number 30
Driver c:\winnt\system32\drivers\q57w2k.sys (77776, 2.75.0.0)

Name [00000002] RAS Async Adapter
Adapter TypeNot Available
Product Name RAS Async Adapter
Installed True
PNP Device ID Not Available
Last Reset 3/3/2003 5:56:57 PM
Index 2
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Not Available

Name [00000003] WAN Miniport (L2TP)
Adapter TypeNot Available
Product Name WAN Miniport (L2TP)
Installed True
PNP Device ID ROOT\MS_L2TPMINIPOINT\0000
Last Reset 3/3/2003 5:56:57 PM
Index 3
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Rasl2tp

Driver c:\winnt\system32\drivers\rasl2tp.sys (52112, 5.00.2195.4052)

Name [00000004] WAN Miniport (PPTP)
 Adapter Type Wide Area Network (WAN)
 Product Name WAN Miniport (PPTP)
 Installed True
 PNP Device ID ROOT\MS_PPTP\MINIPOINT\0000
 Last Reset 3/3/2003 5:56:57 PM
 Index 4
 Service Name PptpMiniport
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 50:50:54:50:30:30
 Service Name PptpMiniport
 Driver c:\winnt\system32\drivers\rasptp.sys (47888, 5.00.2195.4080)

Name [00000005] Direct Parallel
 Adapter Type Not Available
 Product Name Direct Parallel
 Installed True
 PNP Device ID ROOT\MS_PT\IMINIPORT\0000
 Last Reset 3/3/2003 5:56:57 PM
 Index 5
 Service Name Raspti
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled False
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Service Name Raspti
 Driver c:\winnt\system32\drivers\raspti.sys (16880, 5.00.2146.1)

Name [00000006] WAN Miniport (IP)
 Adapter Type Not Available
 Product Name WAN Miniport (IP)
 Installed True
 PNP Device ID ROOT\MS_NDISWANIP\0000
 Last Reset 3/3/2003 5:56:57 PM
 Index 6
 Service Name NdisWan
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled False

DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Service Name NdisWan
 Driver c:\winnt\system32\drivers\ndiswan.sys (93104, 5.00.2195.5241)

[Protocol]

Item	Value
Name	MSAFD Tcpip [TCP/IP]
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	16 bytes
MaximumMessageSize	0 bytes
MessageOriented	False
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	True
SupportsGracefulClosing	True
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD Tcpip [UDP/IP]
ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	16 bytes
MaximumMessageSize	65467 bytes
MessageOriented	True
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	True

Name	RSVP UDP Service Provider
ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	16 bytes
MaximumMessageSize	65467 bytes

MessageAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	True
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	True

Name	RSVP TCP Service Provider
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	16 bytes
MaximumMessageSize	0 bytes
MessageOriented	False
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	True
SupportsExpeditedData	True
SupportsGracefulClosing	True
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{E098C77C-5847-452B-9532-634577082B5F}] SEQUENCEPACKET 0
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{E098C77C-5847-452B-9532-634577082B5F}] DATAGRAM 0
ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False

MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{BC5C750E-3D14-4D5C-ABF4-758E2F06E257}] SEQUENCEPACKET 1
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{BC5C750E-3D14-4D5C-ABF4-758E2F06E257}] DATAGRAM 1
ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{85747BE3-880F-4891-AC32-FC3254971B8}] SEQUENCEPACKET 2
ConnectionlessService	True
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{85747BE3-880F-4891-AC32-FC3254971B8}] DATAGRAM 2
ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{2DF65DEB-EE95-4C61-B3D2-D2C0E995E012}] SEQUENCEPACKET 3
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False

SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{2DF65DEB-EE95-4C61-B3D2-D2C0E995E012}] DATAGRAM 3
ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

[WinSock]

Item	Value
File	c:\winnt\system32\winsock.dll
Version	3.10
Size	2.80 KB (2,864 bytes)
File	c:\winnt\system32\wssock32.dll
Version	5.00.2195.4874
Size	21.27 KB (21,776 bytes)

[Ports]

[Following are sub-categories of this main category]

[Serial]

Item	Value
Name	COM1
Status	OK
PNP Device ID	ACPI\PNP0501\0
Maximum Input Buffer Size	0
Maximum Output Buffer Size	False
Settable Baud Rate	True
Settable Data Bits	True
Settable Flow Control	True
Settable Parity	True
Settable Parity Check	True
Settable Stop Bits	True
Settable RLSD	True

Supports RLS D True
 Supports 16 Bit Mode False
 Supports Special Characters False
 Baud Rate 9600
 Bits/Byte 8
 Stop Bits 1
 Parity None
 Busy 0
 Abort Read/Write on Error 0
 Binary Mode Enabled -1
 Continue XMit on XOff 0
 CTS Outflow Control 0
 Discard NULL Bytes 0
 DSR Outflow Control 0
 DSR Sensitivity 0
 DTR Flow Control Type Enable
 EOF Character 0
 Error Replace Character 0
 Error Replacement Enabled 0
 Event Character 0
 Parity Check Enabled 0
 RTS Flow Control Type Enable
 XOff Character 19
 XOffXMit Threshold 512
 XOn Character 17
 XOnXMit Threshold 2048
 XOnXOff InFlow Control 0
 XOnXOff OutFlow Control 0
 IRQ Number 4
 I/O Port 0x03F8-0x03FF
 Driver c:\winnt\system32\drivers\serial.sys (62512, 5.00.2195.5080)

[Parallel]

Item Value
 No parallel port information

[Storage]

[Following are sub-categories of this main category]

[Drives]

Item Value
 Drive A:
 Description 3 1/2 Inch Floppy Drive
 Drive C:
 Description Local Fixed Disk
 Compressed False
 File System NTFS

Free Space 39.98 GB (38,465,572,984 bytes)
 Volume Name
 Volume Serial Number 90983CE8
 Partition Disk #0, Partition #0
 Partition Size 33.91 GB (36,406,378,496 bytes)
 Starting Offset 16384 bytes
 Drive Description Disk drive
 Drive Manufacturer (Standard disk drives)
 Drive Model COMPAQ LOGICAL VOLUME SCSI Disk Device
 Drive BytesPerSector 512
 Drive MediaLoaded True
 Drive MediaType Fixed hard disk media
 Drive Partitions 1
 Drive SCSI Bus 0
 Drive SCSI Logical Unit 0
 Drive SCSI Port 2
 Drive SCSI Target ID 4
 Drive SectorsPerTrack 32
 Drive Size 36414750720 bytes
 Drive TotalCylinders 8716
 Drive TotalSectors 71122560
 Drive TotalTracks 2222580
 Drive TracksPerCylinder 255

[SCSI]

Item Value
 Name Compaq Smart Array 5i
 Caption Compaq Smart Array 5i
 Driver cpqcissm
 Status OK
 PNP Device ID PCI\VEN_0E11&DEV_B178&SUBSYS_40800E11&REV_01\3&267A616A&0&20
 Device ID PCI\VEN_0E11&DEV_B178&SUBSYS_40800E11&REV_01\3&267A616A&0&20
 Device Map Not Available
 Index Not Available
 Max Number Controlled Not Available
 IRQ Number 31
 I/O Port 0x2800-0x28FF
 Driver c:\winnt\system32\drivers\cpqcissm.sys (14992, 5.40.2.0)
 Name QLogic QLA23xx PCI Fibre Channel Adapter
 Caption QLogic QLA23xx PCI Fibre Channel Adapter
 Driver ql2300
 Status OK
 PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&13C0B0C5&0&08

Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&13C0B0C5&0&08
 Description Not Available
 Index Not Available
 Max Number Controlled Not Available
 IRQ Number 28
 I/O Port 0x3000-0x30FF
 Driver c:\winnt\system32\drivers\ql2300.sys (442386, 8.2.1 Beta 2 (W2K VI))

[Printing]

Name Port Name Server Name
 No printing information

[Problem Devices]

Device PNP Device ID Error Code
 No Problem Devices

[USB]

Device PNP Device ID
 Standard OpenHCD USB Host Controller
 PCI\VEN_1166&DEV_0220&SUBSYS_02201166&REV_05\3&267A616A&0&7A
 USB Root Hub USB\ROOT_HUB\4&AF5358C&0

[Software Environment]

[Following are sub-categories of this main category]

[Drivers]

Name	Description	File	Type	Started	Start
Mode	State	Status	Error Control	Accept	Pause
Stop					
abiosdsk	Abiosdsk	Not Available	Kernel Driver	False	Disabled
	Stopped	OK	Ignore	False	False
abp480n5	abp480n5	Not Available	Kernel Driver	False	Disabled
	Stopped	OK	Normal	False	False
acpi	Microsoft ACPI Driver	c:\winnt\system32\drivers\acpi.sys	Kernel Driver	True	Running
	Kernel Driver	True	Boot	Running	OK
	Normal	False	True		
acpiec	ACPIEC	c:\winnt\system32\drivers\acpiec.sys	Kernel		
Driver	False	Disabled	Stopped	OK	Normal
	False	False			
adpu160m	adpu160m	c:\winnt\system32\drivers\adpu160m.sys	Kernel Driver	True	Running
	Kernel Driver	True	Boot	Running	OK
	Normal	False	True		

afd	AFD Networking Support Environment c:\winnt\system32\drivers\afd.sys	Kernel Driver	True		
	Auto Running OK Normal False				
aha154x	Aha154x Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
aic116x	aic116x Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
aic78u2	aic78u2 Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
aic78xx	aic78xx Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
ami0nt	ami0nt Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
amsint	amsint Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
asc	asc Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
asc3350p	asc3350p Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
asc3550	asc3550 Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
asyncmac	RAS Asynchronous Media Driver c:\winnt\system32\drivers\asyncmac.sys	Kernel Driver	False		
	Manual Stopped OK Normal False				
atapi	Standard IDE/ESDI Hard Disk Controller c:\winnt\system32\drivers\atapi.sys	Kernel Driver	True		
	Boot Running OK Normal False				
atdisk	Atdisk Not Available Kernel Driver	False	Disabled		
	Stopped OK Ignore False				
atirage3	atirage3 c:\winnt\system32\drivers\atimpab.sys	Kernel Driver	True		
Driver	True Manual Running OK Ignore				
atmarpc	ATM ARP Client Protocol c:\winnt\system32\drivers\atmarpc.sys	Kernel Driver	False		
	Manual Stopped OK Normal False				
audstub	Audio Stub Driver c:\winnt\system32\drivers\audstub.sys	Kernel Driver	True		
	Manual Running OK Normal False				
beep	Beep c:\winnt\system32\drivers\beep.sys	Kernel Driver	True		
Driver	True System Running OK Normal				
buslogic	BusLogic Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
cd20xmt	cd20xmt Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
cdaudio	Cdaudio c:\winnt\system32\drivers\cdaudio.sys	Kernel Driver	False	System Stopped OK	Ignore
Driver	False				
cdfs	cdfs c:\winnt\system32\drivers\cdfs.sys	File	False	Running	OK
System Driver	True Disabled				
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				
cpq32fs2	cpq32fs2 c:\winnt\system32\drivers\cpq32fs2.sys	Kernel Driver	True	Boot Running	OK
Driver	False True				
cpqarray	Cpqarray c:\winnt\system32\drivers\cpqarray.sys	Kernel Driver	True	Boot Running	OK
Driver	False True				
cpqarry2	cpqarry2 c:\winnt\system32\drivers\cpqarry2.sys	Kernel Driver	True	Boot Running	OK
Driver	False True				
cpqasm2	hp ProLiant iLO Advanced System Management Controller c:\winnt\system32\drivers\cpqasm2.sys	Kernel Driver	True	Manual Running	OK
	Manual Running OK Normal False				
cpqcidrv	HP Integrated Lights-Out c:\winnt\system32\drivers\cpqcidrv.sys	Kernel Driver	True	Manual Running	OK
	Manual Running OK Normal False				
cpqcisse	CPQCISSE c:\winnt\system32\drivers\cpqcisse.sys	Kernel Driver	True	System Running	OK
Driver	False True				
cpqcissm	cpqcissm c:\winnt\system32\drivers\cpqcissm.sys	Kernel Driver	True	Boot Running	OK
Driver	False True				
cpqfcalm	cpqfcalm Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
cpqfws2e	cpqfws2e Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
cpqteam	Compaq Network Teaming and Configuration c:\winnt\system32\drivers\cpqteam.sys	Kernel Driver	False	Manual Stopped	OK
	Manual Stopped OK Normal False				
dac960nt	dac960nt Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
deckzpsx	deckzpsx Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
dfsdriver	DfsDriver c:\winnt\system32\drivers\dfs.sys	File	True	Boot Running	OK
System Driver	Normal False True				
disk	Disk Driver c:\winnt\system32\drivers\disk.sys	Kernel Driver	True	Boot Running	OK
Driver	False True				
diskperf	Diskperf c:\winnt\system32\drivers\diskperf.sys	Kernel Driver	True	Boot Running	OK
Driver	False				
dmboot	dmboot c:\winnt\system32\drivers\dmboot.sys	Kernel Driver	True	Boot Running	OK
Driver	False Disabled				
dmio	Logical Disk Manager Driver c:\winnt\system32\drivers\dmio.sys	Kernel Driver	True	Boot Running	OK
	True Running OK Normal False				
dmload	dmload c:\winnt\system32\drivers\dmload.sys	Kernel Driver	True	Boot Running	OK
Driver	False True				
efs	EFS c:\winnt\system32\drivers\efs.sys	File	True	Disabled	Running
System Driver	Normal False True				
fastfat	Fastfat c:\winnt\system32\drivers\fastfat.sys	File	True	Disabled	Running
System Driver	Normal False True				
fd16_700	Fd16_700 Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
fdc	Floppy Disk Controller Driver c:\winnt\system32\drivers\fdc.sys	Kernel Driver	True	Manual Running	OK
	Manual Running OK Normal False				
fips	Fips c:\winnt\system32\drivers\fips.sys	Kernel Driver	True	Auto Running	OK
Driver	False True				
fireport	fireport Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
flashpnt	flashpnt Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
flpydisk	Floppy Disk Driver c:\winnt\system32\drivers\flpydisk.sys	Kernel Driver	True	Manual Running	OK
	Manual Running OK Normal False				
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
	Manual Running OK Normal False				
i8042prt	i8042 Keyboard and PS/2 Mouse Port Driver c:\winnt\system32\drivers\i8042prt.sys	Kernel Driver	True	System Running	OK
	True Running OK Normal False				
ini910u	ini910u Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				
intelide	IntelIde Not Available Kernel Driver	False	Disabled		
	Stopped OK Normal False				

ipfilterdriver	IP Traffic Filter Driver	c:\winnt\system32\drivers\ipfltdrv.sys	Kernel Driver	False	False
	Manual	Stopped	OK	Normal	False
ipinip	IP in IP Tunnel Driver	c:\winnt\system32\drivers\ipinip.sys	Kernel Driver	False	OK
	Normal	False	False	Stopped	OK
ipnat	IP Network Address Translator	c:\winnt\system32\drivers\ipnat.sys	Kernel Driver	False	False
	Manual	Stopped	OK	Normal	False
ipsec Driver	IPSEC driver	c:\winnt\system32\drivers\ipsec.sys	Kernel	True	Normal
	True	Manual	Running	OK	Normal
ipsraidn	ipsraidn	Not Available	Kernel Driver	False	Disabled
	Stopped	OK	Normal	False	False
irenum	IR Enumerator Service	c:\winnt\system32\drivers\irenum.sys	Kernel Driver	False	False
	Manual	Stopped	OK	Normal	False
isapnp	PnP ISA/EISA Bus Driver	c:\winnt\system32\drivers\isapnp.sys	Kernel Driver	True	False
	Boot	Running	OK	Critical	False
kbdclass	Keyboard Class Driver	c:\winnt\system32\drivers\kbdclass.sys	Kernel Driver	True	False
	System	Running	OK	Normal	False
ksecdd Driver	KSecDD	c:\winnt\system32\drivers\ksecdd.sys	Kernel	True	Normal
	True	Boot	Running	OK	Normal
lbrtfdc	lbrtfdc	Not Available	Kernel Driver	False	System
	Stopped	OK	Ignore	False	False
lp6nds35	Ip6nds35	Not Available	Kernel Driver	False	Disabled
	Stopped	OK	Normal	False	False
megaide Driver	MegaIDE	c:\winnt\system32\drivers\megaide.sys	Kernel	False	Normal
	False	Disabled	Stopped	OK	Normal
mnmdm Driver	mnmdm	c:\winnt\system32\drivers\mnmdm.sys	Kernel	True	Ignore
	True	System	Running	OK	Ignore
modem Driver	Modem	c:\winnt\system32\drivers\modem.sys	Kernel	False	Ignore
	False	Manual	Stopped	OK	Ignore
mouclass	Mouse Class Driver	c:\winnt\system32\drivers\mouclass.sys	Kernel Driver	True	False
	System	Running	OK	Normal	False
mountmgr	MountMgr	c:\winnt\system32\drivers\mountmgr.sys	Kernel Driver	True	OK
	Normal	False	True	Running	OK

mraid3x	mraid3x	Not Available	Kernel Driver	False	Disabled
mxsmb	mxsmb	c:\winnt\system32\drivers\mxsmb.sys	File	True	False
System Driver	System Driver	True	System	Running	OK
msfs	Msfs	c:\winnt\system32\drivers\msfs.sys	File	True	OK
System Driver	System Driver	True	System	Running	OK
mskssrv	Microsoft Streaming Service Proxy	c:\winnt\system32\drivers\mskssrv.sys	Kernel Driver	False	False
	Manual	Stopped	OK	Normal	False
mspclock	Microsoft Streaming Clock Proxy	c:\winnt\system32\drivers\mspclock.sys	Kernel Driver	False	False
	Manual	Stopped	OK	Normal	False
mspqm	Microsoft Streaming Quality Manager Proxy	c:\winnt\system32\drivers\mspqm.sys	Kernel Driver	False	False
	Manual	Stopped	OK	Normal	False
mup	Mup	c:\winnt\system32\drivers\mup.sys	File	True	OK
System Driver	System Driver	True	Boot	Running	OK
ncrc710	Nrc710	Not Available	Kernel Driver	False	Disabled
	Stopped	OK	Normal	False	False
ndis	NDIS System Driver	c:\winnt\system32\drivers\ndis.sys	Kernel Driver	True	OK
	Normal	False	True	Running	OK
ndistapi	Remote Access NDIS TAPI Driver	c:\winnt\system32\drivers\ndistapi.sys	Kernel Driver	True	False
	Manual	Running	OK	Normal	False
ndiswan	Remote Access NDIS WAN Driver	c:\winnt\system32\drivers\ndiswan.sys	Kernel Driver	True	False
	Manual	Running	OK	Normal	False
ndproxy Driver	NDIS Proxy	c:\winnt\system32\drivers\ndproxy.sys	Kernel	True	Normal
	True	Manual	Running	OK	Normal
netbios	NetBIOS Interface	c:\winnt\system32\drivers\netbios.sys	File System Driver	True	Normal
	True	System	Running	OK	Normal
netbt	NetBios over Tcpip	c:\winnt\system32\drivers\netbt.sys	Kernel Driver	True	OK
	Normal	False	True	Running	OK
netdetect Driver	NetDetect	c:\winnt\system32\drivers\netdetect.sys	Kernel	False	Normal
	False	Manual	Stopped	OK	Normal
npfs	Npfs	c:\winnt\system32\drivers\npfs.sys	File	True	OK
System Driver	System Driver	True	System	Running	OK
	Normal	False	True	Running	OK

ntfs	Ntfs	c:\winnt\system32\drivers\ntfs.sys	File	True	OK
System Driver	System Driver	True	Disabled	Running	OK
null Driver	Null	c:\winnt\system32\drivers\null.sys	Kernel	True	Normal
	Normal	True	System	Running	OK
nwlkflt	IPX Traffic Filter Driver	c:\winnt\system32\drivers\nwlkflt.sys	Kernel Driver	False	False
	Manual	Stopped	OK	Normal	False
nwlkfwfwd	IPX Traffic Forwarder Driver	c:\winnt\system32\drivers\nwlkfwfwd.sys	Kernel Driver	False	False
	Manual	Stopped	OK	Normal	False
openhci	Microsoft USB Open Host Controller Driver	c:\winnt\system32\drivers\openhci.sys	Kernel Driver	True	False
	Manual	Running	OK	Normal	False
parallel Driver	Parallel	c:\winnt\system32\drivers\parallel.sys	Kernel	False	Ignore
	False	Manual	Stopped	OK	Ignore
parport Driver	Parport	c:\winnt\system32\drivers\parport.sys	Kernel	False	Ignore
	False	Manual	Stopped	OK	Ignore
partmgr Driver	PartMgr	c:\winnt\system32\drivers\partmgr.sys	Kernel	True	Normal
	True	Boot	Running	OK	Normal
parvdm Driver	ParVdm	c:\winnt\system32\drivers\parvdm.sys	Kernel	False	Ignore
	False	Manual	Stopped	OK	Ignore
pci	PCI Bus Driver	c:\winnt\system32\drivers\pci.sys	Kernel Driver	True	OK
	Critical	False	True	Running	OK
pcidump	PCIDump	Not Available	Kernel Driver	False	System
	Stopped	OK	Ignore	False	False
pciide Driver	PCIIde	c:\winnt\system32\drivers\pciide.sys	Kernel	True	Normal
	True	Boot	Running	OK	Normal
pcmcia Driver	Pcmcia	c:\winnt\system32\drivers\pcmcia.sys	Kernel	False	Normal
	False	Disabled	Stopped	OK	Normal
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
pdframe	PDFRAME	Not Available	Kernel Driver	False	Manual
	Stopped	OK	Ignore	False	False
pptpminiport	WAN Miniport (PPTP)	c:\winnt\system32\drivers\rasppptp.sys	Kernel Driver	True	False
	Manual	Running	OK	Normal	False
	True				

ptilink	Direct Parallel Link Driver	c:\winnt\system32\drivers\ptilink.sys	Kernel Driver	True	Manual	Running	OK
q57w2k	Compaq NC7781 Gigabit Server Adapter	c:\winnt\system32\drivers\q57w2k.sys	Kernel Driver	True	Manual	Running	OK
ql1080	ql1080	Not Available	Kernel Driver	False	Disabled		
ql10wnt	Ql10wnt	Not Available	Kernel Driver	False	Disabled		
ql1240	ql1240	Not Available	Kernel Driver	False	Disabled		
ql2100	ql2100	Not Available	Kernel Driver	False	Disabled		
ql2300	ql2300	c:\winnt\system32\drivers\ql2300.sys	Kernel Driver	True	Boot	Running	OK
qlvika	qlvika	c:\winnt\system32\drivers\qlvika.sys	Kernel Driver	True	Auto	Running	OK
rasacd	Remote Access Auto Connection Driver	c:\winnt\system32\drivers\rasacd.sys	Kernel Driver	True	System	Running	OK
rasl2tp	WAN Miniport (L2TP)	c:\winnt\system32\drivers\rasl2tp.sys	Kernel Driver	True	Manual	Running	OK
raspti	Direct Parallel	c:\winnt\system32\drivers\raspti.sys	Kernel Driver	True	Manual	Running	OK
rca	Microsoft Streaming Network Raw Channel Access	c:\winnt\system32\drivers\rca.sys	Kernel Driver	False	Manual	Stopped	OK
rdbss	Rdbss	c:\winnt\system32\drivers\rdbss.sys	File System Driver	True	System	Running	OK
rdpdr	Terminal Server Device Redirector Driver	c:\winnt\system32\drivers\rdpdr.sys	Kernel Driver	True	Manual	Running	OK
rdpwd	RDPWD	c:\winnt\system32\drivers\rdpwd.sys	Kernel Driver	True	Manual	Running	OK
redbook	Digital CD Audio Playback Filter Driver	c:\winnt\system32\drivers\redbook.sys	Kernel Driver	False	System	Stopped	OK
serenum	Serenum Filter Driver	c:\winnt\system32\drivers\serenum.sys	Kernel Driver	True	Manual	Running	OK
serial	Serial port driver	c:\winnt\system32\drivers\serial.sys	Kernel Driver	True	Manual	Running	OK
sfloppy	Sfloppy	c:\winnt\system32\drivers\sfloppy.sys	Kernel Driver	False	System	Stopped	OK
sglfb	sglfb	Not Available	Kernel Driver	False	System		
simbad	Simbad	Not Available	Kernel Driver	False	Disabled		
sparrow	Sparrow	Not Available	Kernel Driver	False	Disabled		
spud	Special Purpose Utility Driver	c:\winnt\system32\drivers\spud.sys	Kernel Driver	True	Manual	Running	OK
srv	Srv	c:\winnt\system32\drivers\srv.sys	File System Driver	True	Manual	Running	OK
swenum	Software Bus Driver	c:\winnt\system32\drivers\swenum.sys	Kernel Driver	True	Manual	Running	OK
symc810	symc810	c:\winnt\system32\drivers\symc810.sys	Kernel Driver	True	Boot	Running	OK
symc8xx	symc8xx	c:\winnt\system32\drivers\symc8xx.sys	Kernel Driver	True	Boot	Running	OK
sym_hi	sym_hi	c:\winnt\system32\drivers\sym_hi.sys	Kernel Driver	True	Boot	Running	OK
sysmgmt	hp ProLiant System Management Interface Driver	c:\winnt\system32\drivers\sysmgmt.sys	Kernel Driver	True	Manual	Running	OK
tcpip	TCP/IP Protocol Driver	c:\winnt\system32\drivers\tcpip.sys	Kernel Driver	True	System	Running	OK
tdasync	TDASync	c:\winnt\system32\drivers\tdasync.sys	Kernel Driver	False	Manual	Stopped	OK
tdipx	TDIPX	c:\winnt\system32\drivers\tdipx.sys	Kernel Driver	False	Manual	Stopped	OK
tdnetb	TDNETB	c:\winnt\system32\drivers\tdnetb.sys	Kernel Driver	False	Manual	Stopped	OK
tdpipe	TDPIPE	c:\winnt\system32\drivers\tdpipe.sys	Kernel Driver	False	Manual	Stopped	OK
tdspix	TDSPX	c:\winnt\system32\drivers\tdspix.sys	Kernel Driver	False	Manual	Stopped	OK
tdtcp	TDTCP	c:\winnt\system32\drivers\tdtcp.sys	Kernel Driver	True	Manual	Running	OK
termdd	Terminal Device Driver	c:\winnt\system32\drivers\termdd.sys	Kernel Driver	True	Auto	Running	OK
tga	tga	Not Available	Kernel Driver	False	System		
udfs	Udfs	c:\winnt\system32\drivers\udfs.sys	File System Driver	False	Disabled	Stopped	OK
ultra66	ultra66	Not Available	Kernel Driver	False	Disabled		
update	Microcode Update Driver	c:\winnt\system32\drivers\update.sys	Kernel Driver	True	Manual	Running	OK
usbhub	Microsoft USB Standard Hub Driver	c:\winnt\system32\drivers\usbhub.sys	Kernel Driver	True	Manual	Running	OK
vgasave	VgaSave	c:\winnt\system32\drivers\vga.sys	Kernel Driver	True	System	Running	OK
wanarp	Remote Access IP ARP Driver	c:\winnt\system32\drivers\wanarp.sys	Kernel Driver	True	Manual	Running	OK
wdica	WDICA	Not Available	Kernel Driver	False	Manual	Stopped	OK
[Environment Variables]							
Variable	Value	User Name					
ComSpec	%SystemRoot%\system32\cmd.exe	<SYSTEM>					
Os2LibPath	%SystemRoot%\system32\os2\dll;	<SYSTEM>					
Path	%SystemRoot%\system32;%SystemRoot%\System32\Wbem;C:\Program Files\Microsoft SQL Server\80\Tools\BINN	<SYSTEM>					
windir	%SystemRoot%\	<SYSTEM>					
OS	Windows_NT	<SYSTEM>					
PROCESSOR_ARCHITECTURE	x86	<SYSTEM>					
PROCESSOR_LEVEL	15	<SYSTEM>					
PROCESSOR_IDENTIFIER	x86 Family 15 Model 2 Stepping 7, GenuineIntel	<SYSTEM>					
PROCESSOR_REVISION	0207	<SYSTEM>					

NUMBER_OF_PROCESSORS 4 <SYSTEM>
 PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
 <SYSTEM>
 TEMP %SystemRoot%\TEMP <SYSTEM>
 TMP %SystemRoot%\TEMP <SYSTEM>
 TEMP %USERPROFILE%\Local Settings\Temp W11\Administrator
 TMP %USERPROFILE%\Local Settings\Temp W11\Administrator

[Jobs]

[Following are sub-categories of this main category]

[Print]

Document	Size	Owner	Notify	Status	Time
Submitted	Start Time	Until Time	Elapsed Time	Pages Printed	
	Job ID	Priority	Parameters	Driver Name	Print
Processor	Host Print Queue	Data Type	Name		
No print jobs					

[Network Connections]

Local Name	Remote Name	Type	Status	User
Name				
E:	\\ssd\\lab\\d\$ Disk	OK		

[Running Tasks]

Name	Path	Process ID	Priority	Min Working Set
	Max Working Set		Start Time	Version
	File Date			Size
system idle process	Not Available	0	0	Not
Available	Not Available	Not Available	Unknown	Unknown
system	Not Available	8	0	1413120
	Not Available	Unknown	Unknown	Unknown
smss.exe	c:\winnt\system32\smss.exe	204800	1413120	11
		3/4/2003 1:57:36 AM		
	5.00.2195.5382	44.77 KB (45,840 bytes)		
	12/31/1979 4:00:00 PM			
csrss.exe	Not Available	264	13	Not Available
Available	3/4/2003 1:57:39 AM	Unknown	Unknown	Unknown
winlogon.exe	c:\winnt\system32\winlogon.exe	204800	1413120	13
		3/4/2003 1:57:39 AM		
	5.00.2195.5386	174.77 KB (178,960 bytes)	4/2/2003	
1:16:46 PM				
services.exe	c:\winnt\system32\services.exe	204800	1413120	9
		3/4/2003 1:57:40 AM		
	5.00.2195.3940	86.77 KB (88,848 bytes)		
	12/31/1979 4:00:00 PM			

lsass.exe	c:\winnt\system32\lsass.exe	204800	1413120	3/4/2003 1:57:40 AM	324	9
		5.00.2195.5430	32.77 KB (33,552 bytes)			
termsrv.exe	c:\winnt\system32\termsrv.exe	204800	1413120	3/4/2003 1:57:40 AM	432	10
		5.00.2195.5276	138.77 KB (142,096 bytes)	4/2/2003		
1:16:45 PM						
svchost.exe	c:\winnt\system32\svchost.exe	204800	1413120	3/4/2003 1:57:41 AM	536	8
		5.00.2134.1	7.77 KB (7,952 bytes)	12/31/1979 4:00:00		
PM						
spoolsv.exe	c:\winnt\system32\spoolsv.exe	204800	1413120	3/4/2003 1:57:41 AM	568	8
		5.00.2195.4299	44.27 KB (45,328 bytes)	4/2/2003		
3:27:50 AM						
msdtc.exe	c:\winnt\system32\msdtc.exe	204800	1413120	3/4/2003 1:57:41 AM	596	8
		1999.9.3421.3	6.77 KB (6,928 bytes)	4/2/2003		
3:33:33 AM						
cpqrcmc.exe	c:\winnt\system32\cpqrcmc.exe	204800	1413120	3/4/2003 1:57:42 AM	760	8
		96.27 KB (98,576 bytes)	2/7/2001 4:40:24 PM	5.0.2.0		
svchost.exe	c:\winnt\system32\svchost.exe	204800	1413120	3/4/2003 1:57:42 AM	780	8
		5.00.2134.1	7.77 KB (7,952 bytes)	12/31/1979 4:00:00		
PM						
llssrv.exe	c:\winnt\system32\llssrv.exe	204800	1413120	3/4/2003 1:57:42 AM	800	9
		5.00.2195.4907	81.27 KB (83,216 bytes)			
		7/22/2002 1:05:04 PM				
regsvc.exe	c:\winnt\system32\regsvc.exe	204800	1413120	3/4/2003 1:57:43 AM	884	8
		5.00.2195.3649	65.27 KB (66,832 bytes)	4/2/2003		
1:16:42 PM						
rsys.exe	Not Available	904	8	Not Available	Not	
Available	3/4/2003 1:57:43 AM	Unknown	Unknown	Unknown	Unknown	
mstask.exe	c:\winnt\system32\mstask.exe	204800	1413120	3/4/2003 1:57:43 AM	916	8
		4.71.2195.1	115.77 KB (118,544 bytes)	4/2/2003 1:16:37 PM		
snmp.exe	c:\winnt\system32\snmp.exe	204800	1413120	3/4/2003 1:57:43 AM	940	8
		5.00.2195.5080	29.77 KB (30,480 bytes)	4/2/2003		
3:33:32 AM						
winmgmt.exe	c:\winnt\system32\wbem\winmgmt.exe	204800	1413120	3/4/2003 1:57:43 AM	1012	8
		1.50.1085.0070	192.08 KB (196,685 bytes)	4/2/2003		
1:16:50 PM						
svchost.exe	c:\winnt\system32\svchost.exe	204800	1413120	3/4/2003 1:57:43 AM	1028	8
		5.00.2134.1	7.77 KB (7,952 bytes)	12/31/1979 4:00:00		
PM						

inetinfo.exe	c:\winnt\system32\inetrv\inetinfo.exe	204800	1413120	3/4/2003 1:57:43 AM	1040	8
		5.00.2195.5430	32.77 KB (33,552 bytes)	4/2/2003 1:16:01 PM		
sysdown.exe	c:\winnt\system32\sysdown.exe	204800	1413120	3/4/2003 1:57:43 AM	1056	8
		5.26.2195.0	34.77 KB (35,600 bytes)	4/2/2003 11:42:10 AM		
dfssvc.exe	c:\winnt\system32\dfssvc.exe	204800	1413120	3/4/2003 1:57:44 AM	1112	8
		5.00.2195.3649	88.27 KB (90,384 bytes)	4/2/2003		
1:16:28 PM						
svchost.exe	c:\winnt\system32\svchost.exe	204800	1413120	3/4/2003 1:57:56 AM	1452	8
		5.00.2134.1	7.77 KB (7,952 bytes)	12/31/1979 4:00:00		
PM						
logon.scr	c:\winnt\system32\logon.scr	204800	1413120	3/4/2003 2:37:56 AM	456	4
		5.00.2195.5305	127.77 KB (130,832 bytes)	4/2/2003		
1:16:33 PM						
csrss.exe	Not Available	1536	13	Not Available	Not	
Available	3/4/2003 6:39:30 PM	Unknown	Unknown	Unknown	Unknown	
winlogon.exe	c:\winnt\system32\winlogon.exe	204800	1413120	3/4/2003 6:39:30 PM	1564	13
		5.00.2195.5386	174.77 KB (178,960 bytes)	4/2/2003		
1:16:46 PM						
rdpclip.exe	c:\winnt\system32\rdpclip.exe	204800	1413120	3/4/2003 6:39:35 PM	1588	8
		5.00.2174.1	39.77 KB (40,720 bytes)	4/2/2003 3:33:34 AM		
explorer.exe	c:\winnt\explorer.exe	1413120	3/4/2003 6:39:35 PM	5.00.3502.5321	204800	
		237.27 KB (242,960 bytes)	4/2/2003 1:16:47 PM			
cpqteam.exe	c:\winnt\system32\cpqteam.exe	204800	1413120	3/4/2003 6:39:35 PM	1388	8
		7.10.710.18	84.09 KB (86,112 bytes)	6/12/2002 8:58:50 AM		
mmc.exe	c:\winnt\system32\mmc.exe	204800	1413120	3/4/2003 6:39:39 PM	1608	8
		5.00.2195.4933	589.27 KB (603,408 bytes)	4/2/2003		
1:16:33 PM						
rsvp.exe	c:\winnt\system32\rsvp.exe	204800	1413120	3/4/2003 6:40:58 PM	1788	8
		5.00.2167.1	172.77 KB (176,912 bytes)	12/31/1979 4:00:00		
PM						

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
traffic.dll	5.00.2139.1	30.77 KB (31,504 bytes)	12/31/1979 4:00:00		
PM					
rsvp.exe	5.00.2167.1	172.77 KB (176,912 bytes)	12/31/1979 4:00:00		
PM					
wbemprox.dll	1.50.1085.0045	40.08 KB (41,040 bytes)	4/2/2003		
1:16:50 PM					
mlang.dll	5.00.3315.3727	509.77 KB (522,000 bytes)	4/2/2003		
1:16:33 PM					

cabinet.dll	5.00.2147.1	54.77 KB (56,080 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation	c:\winnt\system32\cabinet.dll
msinfo32.dll	5.00.2195.4601	312.27 KB (319,760 bytes)	4/2/2003 1:16:52 PM	Microsoft Corporation	c:\program files\common files\microsoft shared\msinfo\msinfo32.dll
mmcndmgr.dll	5.00.2195.5352	816.27 KB (835,856 bytes)	4/2/2003 1:16:33 PM	Microsoft Corporation	c:\winnt\system32\mmcndmgr.dll
mmc.exe	5.00.2195.4933	589.27 KB (603,408 bytes)	4/2/2003 1:16:33 PM	Microsoft Corporation	c:\winnt\system32\mmc.exe
cpqteam.exe	7.10.710.18	84.09 KB (86,112 bytes)	6/12/2002 8:58:50 AM	Compaq Computer Corporation	c:\winnt\system32\cpqteam.exe
shdoclc.dll	5.00.3502.5039	324.50 KB (332,288 bytes)	4/2/2003 1:16:43 PM	Microsoft Corporation	c:\winnt\system32\shdoclc.dll
linkinfo.dll	5.00.2134.1	15.77 KB (16,144 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation	c:\winnt\system32\linkinfo.dll
powrprof.dll	5.00.3502.5305	13.27 KB (13,584 bytes)	4/2/2003 1:16:41 PM	Microsoft Corporation	c:\winnt\system32\powrprof.dll
batmeter.dll	5.00.3502.5305	20.27 KB (20,752 bytes)	4/2/2003 1:16:25 PM	Microsoft Corporation	c:\winnt\system32\batmeter.dll
stobject.dll	5.00.2195.4455	79.27 KB (81,168 bytes)	4/2/2003 1:16:44 PM	Microsoft Corporation	c:\winnt\system32\stobject.dll
msi.dll	2.0.2600.1	1.90 MB (1,991,168 bytes)	4/2/2003 1:16:35 PM	Microsoft Corporation	c:\winnt\system32\msi.dll
webcheck.dll	5.00.3315.3727	250.77 KB (256,784 bytes)	4/2/2003 1:16:46 PM	Microsoft Corporation	c:\winnt\system32\webcheck.dll
ntshrui.dll	5.00.2134.1	46.77 KB (47,888 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation	c:\winnt\system32\ntshrui.dll
mydocs.dll	5.00.3315.4065	55.27 KB (56,592 bytes)	4/2/2003 1:16:38 PM	Microsoft Corporation	c:\winnt\system32\mydocs.dll
browseui.dll	5.00.3502.4373	791.27 KB (810,256 bytes)	4/2/2003 1:16:25 PM	Microsoft Corporation	c:\winnt\system32\browseui.dll
shdocvw.dll	5.00.3502.5400	1.05 MB (1,105,168 bytes)	4/2/2003 1:16:43 PM	Microsoft Corporation	c:\winnt\system32\shdocvw.dll
explorer.exe	5.00.3502.5321	237.27 KB (242,960 bytes)	4/2/2003 1:16:47 PM	Microsoft Corporation	c:\winnt\explorer.exe
rdpclip.exe	5.00.2174.1	39.77 KB (40,720 bytes)	4/2/2003 3:33:34 AM	Microsoft Corporation	c:\winnt\system32\rdpclip.exe
mscms.dll	5.00.2180.1	68.27 KB (69,904 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation	c:\winnt\system32\mscms.dll
printui.dll	5.00.2195.5212	372.27 KB (381,200 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation	c:\winnt\system32\printui.dll
netmsg.dll	5.00.2137.1	152.50 KB (156,160 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation	c:\winnt\system32\netmsg.dll
comdlg32.dll	5.00.3315.3727	221.27 KB (226,576 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation	c:\winnt\system32\comdlg32.dll
netui2.dll	5.00.2134.1	280.27 KB (286,992 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation	c:\winnt\system32\netui2.dll

mprui.dll	5.00.2195.4874	54.77 KB (56,080 bytes)	4/2/2003 1:16:34 PM	Microsoft Corporation	c:\winnt\system32\mprui.dll
csui.dll	5.00.2195.5305	127.77 KB (130,832 bytes)	4/2/2003 1:16:33 PM	Microsoft Corporation	c:\winnt\system32\csui.dll
logon.scr	5.00.2195.5305	127.77 KB (130,832 bytes)	4/2/2003 1:16:33 PM	Microsoft Corporation	c:\winnt\system32\logon.scr
tapisrv.dll	5.00.2195.5227	169.27 KB (173,328 bytes)	4/2/2003 1:16:45 PM	Microsoft Corporation	c:\winnt\system32\tapisrv.dll
dfssvc.exe	5.00.2195.3649	88.27 KB (90,384 bytes)	4/2/2003 1:16:28 PM	Microsoft Corporation	c:\winnt\system32\dfssvc.exe
sysdown.exe	5.26.2195.0	34.77 KB (35,600 bytes)	4/2/2003 11:42:10 AM	Compaq Computer Corporation	c:\winnt\system32\sysdown.exe
ntfsdrv.dll	5.00.0984	37.27 KB (38,160 bytes)	4/2/2003 1:17:02 PM	Microsoft Corporation	c:\winnt\system32\ntfsdrv.dll
aqueue.dll	5.00.0984	313.77 KB (321,296 bytes)	4/2/2003 1:16:55 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\aqueue.dll
httpext.dll	5.00.0984	240.27 KB (246,032 bytes)	4/2/2003 1:17:01 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\httpext.dll
md5filt.dll	5.00.0984	32.77 KB (33,552 bytes)	4/2/2003 1:17:02 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\md5filt.dll
gzip.dll	5.00.0984	30.27 KB (30,992 bytes)	4/2/2003 1:17:01 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\gzip.dll
compfilt.dll	5.00.0984	22.77 KB (23,312 bytes)	4/2/2003 1:17:01 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\compfilt.dll
seo.dll	5.00.0984	229.27 KB (234,768 bytes)	4/2/2003 1:17:02 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\seo.dll
sspfilt.dll	5.00.0984	42.77 KB (43,792 bytes)	4/2/2003 1:17:02 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\sspfilt.dll
iscomlog.dll	5.00.0984	24.27 KB (24,848 bytes)	4/2/2003 1:17:02 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\iscomlog.dll
lonsint.dll	5.00.0984	11.77 KB (12,048 bytes)	4/2/2003 1:17:02 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\lonsint.dll
inetsloc.dll	5.00.0984	20.27 KB (20,752 bytes)	4/2/2003 1:16:31 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\inetsloc.dll
w3svc.dll	5.00.0984	335.27 KB (343,312 bytes)	4/2/2003 1:17:02 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\w3svc.dll
staxmem.dll	5.00.0984	8.27 KB (8,464 bytes)	4/2/2003 1:16:44 PM	Microsoft Corporation	c:\winnt\system32\staxmem.dll
exstrace.dll	5.00.0984	13.77 KB (14,096 bytes)	4/2/2003 3:33:45 AM	Microsoft Corporation	c:\winnt\system32\exstrace.dll
rwnh.dll	5.00.0984	10.77 KB (11,024 bytes)	4/2/2003 1:16:42 PM	Microsoft Corporation	c:\winnt\system32\rwnh.dll
fcachdll.dll	5.00.0984	43.77 KB (44,816 bytes)	4/2/2003 1:16:30 PM	Microsoft Corporation	c:\winnt\system32\fcachdll.dll
iisfecnv.dll	5.00.0984	7.27 KB (7,440 bytes)	4/2/2003 3:33:45 AM	Microsoft Corporation	c:\winnt\system32\inetsrv\iisfecnv.dll
isatq.dll	5.00.0984	60.77 KB (62,224 bytes)	4/2/2003 1:17:01 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\isatq.dll

infocomm.dll	5.00.0984	240.77 KB (246,544 bytes)	4/2/2003 1:17:01 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\infocomm.dll
smtpsvc.dll	5.00.0984	128.27 KB (130,816 bytes)	4/2/2003 1:17:02 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\smtpsvc.dll
security.dll	5.00.2154.1	5.77 KB (5,904 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation	c:\winnt\system32\security.dll
svcxext.dll	5.00.0984	39.77 KB (40,720 bytes)	4/2/2003 1:17:02 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\svcxext.dll
admexs.dll	5.00.0984	27.77 KB (28,432 bytes)	4/2/2003 1:17:01 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\admexs.dll
wamreg.dll	5.00.0984	45.77 KB (46,864 bytes)	4/2/2003 1:17:02 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\wamreg.dll
metadata.dll	5.00.0984	68.77 KB (70,416 bytes)	4/2/2003 1:17:02 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\metadata.dll
iismap.dll	5.00.0984	55.77 KB (57,104 bytes)	4/2/2003 1:16:31 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\iismap.dll
nsepm.dll	5.00.0984	43.27 KB (44,304 bytes)	4/2/2003 1:17:02 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\nsepm.dll
admwxprox.dll	5.00.0984	31.77 KB (32,528 bytes)	4/2/2003 3:33:45 AM	Microsoft Corporation	c:\winnt\system32\inetsrv\admwxprox.dll
coadmin.dll	5.00.0984	39.77 KB (40,720 bytes)	4/2/2003 1:17:01 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\coadmin.dll
iisadmin.dll	5.00.0984	15.27 KB (15,632 bytes)	4/2/2003 1:17:01 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\iisadmin.dll
rpcref.dll	5.00.0984	4.27 KB (4,368 bytes)	4/2/2003 1:17:02 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\rpcref.dll
inetinfo.exe	5.00.0984	14.27 KB (14,608 bytes)	4/2/2003 1:17:01 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\inetinfo.exe
winhttp.dll	5.1.2600.1039 (xpsp1.020511-1800)	303.00 KB (310,272 bytes)	4/2/2003 1:17:00 PM	Microsoft Corporation	c:\winnt\system32\winhttp.dll
wininet.dll	5.00.3502.4619	450.77 KB (461,584 bytes)	4/2/2003 1:16:46 PM	Microsoft Corporation	c:\winnt\system32\wininet.dll
utilidl.dll	5.00.2153.1	25.77 KB (26,384 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation	c:\winnt\system32\utilidl.dll
wtsapi32.dll	5.00.2134.1	14.27 KB (14,608 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation	c:\winnt\system32\wtsapi32.dll
advpack.dll	5.00.3502.4373	86.77 KB (88,848 bytes)	4/2/2003 1:16:24 PM	Microsoft Corporation	c:\winnt\system32\advpack.dll
wuaueng.dll	5.4.3628.1	182.50 KB (186,880 bytes)	4/2/2003 1:17:00 PM	Microsoft Corporation	c:\winnt\system32\wuaueng.dll
wuauerv.dll	5.4.3628.1	8.50 KB (8,704 bytes)	4/2/2003 1:17:00 PM	Microsoft Corporation	c:\winnt\system32\wuauerv.dll

netui1.dll 5.00.2134.1 210.27 KB (215,312 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\netui1.dll
 netui0.dll 5.00.2195.4874 70.77 KB (72,464 bytes) 4/2/2003 1:16:39 PM Microsoft Corporation c:\winnt\system32\netui0.dll
 ntlanman.dll 5.00.2195.5428 35.27 KB (36,112 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\ntlanman.dll
 wshnetbs.dll 5.00.2134.1 7.77 KB (7,952 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\wshnetbs.dll
 ntmarta.dll 5.00.2195.4836 99.77 KB (102,160 bytes) 4/2/2003 1:16:39 PM Microsoft Corporation c:\winnt\system32\ntmarta.dll
 provthrd.dll 1.50.1085.0000 68.07 KB (69,708 bytes) 4/2/2003 11:35:50 AM Microsoft Corporation c:\winnt\system32\wbem\provthrd.dll
 ntevt.dll 1.50.1085.0072 192.06 KB (196,671 bytes) 4/2/2003 1:16:50 PM Microsoft Corporation c:\winnt\system32\wbem\ntevt.dll
 psapi.dll 5.00.2134.1 28.27 KB (28,944 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\psapi.dll
 framedyn.dll 1.50.1085.0076 164.07 KB (168,009 bytes) 4/2/2003 1:16:49 PM Microsoft Corporation c:\winnt\system32\wbem\framedyn.dll
 cimwin32.dll 1.50.1085.0073 1.04 MB (1,085,520 bytes) 4/2/2003 1:16:49 PM Microsoft Corporation c:\winnt\system32\wbem\cimwin32.dll
 wbemsvc.dll 1.50.1085.0007 40.07 KB (41,036 bytes) 4/2/2003 1:16:50 PM Microsoft Corporation c:\winnt\system32\wbem\wbemsvc.dll
 wbemess.dll 1.50.1085.0074 364.07 KB (372,804 bytes) 4/2/2003 1:16:50 PM Microsoft Corporation c:\winnt\system32\wbem\wbemess.dll
 fastprox.dll 1.50.1085.0056 144.08 KB (147,536 bytes) 4/2/2003 1:16:49 PM Microsoft Corporation c:\winnt\system32\wbem\fastprox.dll
 wbemcore.dll 1.50.1085.0085 628.07 KB (643,146 bytes) 4/2/2003 1:16:50 PM Microsoft Corporation c:\winnt\system32\wbem\wbemcore.dll
 wbemcomn.dll 1.50.1085.0077 692.07 KB (708,675 bytes) 4/2/2003 1:16:50 PM Microsoft Corporation c:\winnt\system32\wbem\wbemcomn.dll
 winmgmt.exe 1.50.1085.0070 192.08 KB (196,685 bytes) 4/2/2003 1:16:50 PM Microsoft Corporation c:\winnt\system32\wbem\winmgmt.exe
 iisrtd.dll 5.00.0984 119.77 KB (122,640 bytes) 4/2/2003 1:16:31 PM Microsoft Corporation c:\winnt\system32\iisrtd.dll
 infoadmn.dll 5.00.0984 13.27 KB (13,584 bytes) 4/2/2003 1:16:31 PM Microsoft Corporation c:\winnt\system32\infoadmn.dll
 httpmib.dll 5.00.0984 9.27 KB (9,488 bytes) 4/2/2003 1:17:01 PM Microsoft Corporation c:\winnt\system32\inetsrv\httpmib.dll
 rtpxmib.dll 5.00.2168.1 29.77 KB (30,480 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\rtpxmib.dll

btppagnt.dll 5.00.2168.1 13.27 KB (13,584 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\btppagnt.dll
 ospagnt.dll 5.00.2168.1 13.27 KB (13,584 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\ospagnt.dll
 ripagnt.dll 5.00.2168.1 24.27 KB (24,848 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\ripagnt.dll
 perfos.dll 5.00.2155.1 21.27 KB (21,776 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\perfos.dll
 mcastmib.dll 5.00.2168.1 13.27 KB (13,584 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\mcastmib.dll
 igmpagnt.dll 5.00.2168.1 8.77 KB (8,976 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\igmpagnt.dll
 acsmib.dll 5.00.2167.1 11.27 KB (11,536 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\acsmib.dll
 evntagnt.dll 5.00.2195.4874 95.27 KB (97,552 bytes) 4/2/2003 1:16:30 PM Microsoft Corporation c:\winnt\system32\evntagnt.dll
 snmpmib.dll 5.00.2134.1 5.77 KB (5,904 bytes) 4/2/2003 3:33:32 AM Microsoft Corporation c:\winnt\system32\snmpmib.dll
 hostmib.dll 5.00.2195.4874 36.27 KB (37,136 bytes) 4/2/2003 1:16:30 PM Microsoft Corporation c:\winnt\system32\hostmib.dll
 inetmib1.dll 5.00.2195.4874 28.77 KB (29,456 bytes) 4/2/2003 1:16:31 PM Microsoft Corporation c:\winnt\system32\inetmib1.dll
 lmmib2.dll 5.00.2195.4874 29.27 KB (29,968 bytes) 4/2/2003 1:16:33 PM Microsoft Corporation c:\winnt\system32\lmmib2.dll
 snmpapi.dll 5.00.2195.4874 17.27 KB (17,680 bytes) 4/2/2003 1:16:44 PM Microsoft Corporation c:\winnt\system32\snmpapi.dll
 snmp.exe 5.00.2195.5080 29.77 KB (30,480 bytes) 4/2/2003 3:33:32 AM Microsoft Corporation c:\winnt\system32\snmp.exe
 msidle.dll 5.00.2920.0000 6.27 KB (6,416 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\msidle.dll
 mstask.exe 4.71.2195.1 115.77 KB (118,544 bytes) 4/2/2003 1:16:37 PM Microsoft Corporation c:\winnt\system32\mstask.exe
 regsvc.exe 5.00.2195.3649 65.27 KB (66,832 bytes) 4/2/2003 1:16:42 PM Microsoft Corporation c:\winnt\system32\regsvc.exe
 llsrc.dll 5.00.2195.4907 47.77 KB (48,912 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\llsrc.dll
 llssrv.exe 5.00.2195.4907 81.27 KB (83,216 bytes) 7/22/2002 1:05:04 PM Microsoft Corporation c:\winnt\system32\llssrv.exe
 wmi.dll 5.00.2191.1 6.27 KB (6,416 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\wmi.dll
 netshell.dll 5.00.2195.5431 457.77 KB (468,752 bytes) 4/2/2003 1:16:39 PM Microsoft Corporation c:\winnt\system32\netshell.dll
 netman.dll 5.00.2195.5282 89.27 KB (91,408 bytes) 4/2/2003 1:16:39 PM Microsoft Corporation c:\winnt\system32\netman.dll
 comsvcs.dll 2000.2.3497.0 1.37 MB (1,439,504 bytes) 4/2/2003 1:16:26 PM Microsoft Corporation c:\winnt\system32\comsvcs.dll
 ntmsdba.dll 5.00.2195.5279 169.27 KB (173,328 bytes) 4/2/2003 1:16:39 PM Microsoft Corporation c:\winnt\system32\ntmsdba.dll
 rasdlg.dll 5.00.2195.5438 515.77 KB (528,144 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\rasdlg.dll

netcfgx.dll 5.00.2195.4874 534.77 KB (547,600 bytes) 4/2/2003 1:16:42 PM Microsoft Corporation c:\winnt\system32\netcfgx.dll
 rasman.dll 5.00.2195.5436 440.77 KB (453,824 bytes) 4/2/2003 1:16:42 PM Microsoft Corporation c:\winnt\system32\rasmans.dll
 sens.dll 5.00.2163.1 36.77 KB (37,648 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\sens.dll
 ntmssvc.dll 5.00.2195.5254 391.77 KB (401,168 bytes) 4/2/2003 1:16:40 PM Microsoft Corporation c:\winnt\system32\ntmssvc.dll
 es.dll 2000.2.3497.0 225.27 KB (230,672 bytes) 4/2/2003 1:16:29 PM Microsoft Corporation c:\winnt\system32\es.dll
 cpqrcmc.exe 5.0.2.0 96.27 KB (98,576 bytes) 2/7/2001 4:40:24 PM Compaq c:\winnt\system32\cpqrcmc.exe
 mbxoci.dll 2000.2.3497.0 103.77 KB (106,256 bytes) 4/2/2003 1:16:38 PM Microsoft Corporation c:\winnt\system32\mbxoci.dll
 resutils.dll 5.00.2195.5339 39.77 KB (40,720 bytes) 4/2/2003 1:16:42 PM Microsoft Corporation c:\winnt\system32\resutils.dll
 clusapi.dll 5.00.2195.4678 54.27 KB (55,568 bytes) 4/2/2003 1:16:26 PM Microsoft Corporation c:\winnt\system32\clusapi.dll
 msvcp50.dll 5.00.7051 552.50 KB (565,760 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\msvcp50.dll
 xolehlp.dll 1999.9.3421.3 17.27 KB (17,680 bytes) 4/2/2003 3:33:33 AM Microsoft Corporation c:\winnt\system32\xolehlp.dll
 msdtclog.dll 2000.2.3497.0 86.77 KB (88,848 bytes) 4/2/2003 1:16:34 PM Microsoft Corporation c:\winnt\system32\msdtclog.dll
 mbxclu.dll 2000.2.3497.0 51.27 KB (52,496 bytes) 4/2/2003 1:16:38 PM Microsoft Corporation c:\winnt\system32\mbxclu.dll
 msdtcprx.dll 2000.2.3497.0 683.77 KB (700,176 bytes) 4/2/2003 1:16:34 PM Microsoft Corporation c:\winnt\system32\msdtcprx.dll
 txfaux.dll 2000.2.3497.0 383.27 KB (392,464 bytes) 4/2/2003 1:16:45 PM Microsoft Corporation c:\winnt\system32\txfaux.dll
 msdtctm.dll 2000.2.3497.0 1.08 MB (1,128,208 bytes) 4/2/2003 1:16:34 PM Microsoft Corporation c:\winnt\system32\msdtctm.dll
 msdtc.exe 1999.9.3421.3 6.77 KB (6,928 bytes) 4/2/2003 3:33:33 AM Microsoft Corporation c:\winnt\system32\msdtc.exe
 inetpp.dll 5.00.2195.4299 64.27 KB (65,808 bytes) 4/2/2003 1:16:31 PM Microsoft Corporation c:\winnt\system32\inetpp.dll
 win32spl.dll 5.00.2195.5201 92.27 KB (94,480 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\win32spl.dll
 usbmon.dll 5.00.2195.4299 11.27 KB (11,536 bytes) 4/2/2003 1:16:46 PM Microsoft Corporation c:\winnt\system32\usbmon.dll
 tcpmon.dll 5.00.2195.4299 40.77 KB (41,744 bytes) 4/2/2003 1:16:45 PM Microsoft Corporation c:\winnt\system32\tcpmon.dll
 pjimon.dll 5.00.2165.1 12.77 KB (13,072 bytes) 11/30/1999 3:39:36 PM Microsoft Corporation c:\winnt\system32\pjimon.dll
 cnbjmon.dll 5.00.2134.1 43.77 KB (44,816 bytes) 11/30/1999 3:38:48 PM Microsoft Corporation c:\winnt\system32\cnbjmon.dll
 localspl.dll 5.00.2195.5423 250.27 KB (256,272 bytes) 12/31/1979 4:00:00 PM Microsoft Corporation c:\winnt\system32\localspl.dll
 spoolss.dll 5.00.2195.5400 61.77 KB (63,248 bytes) 4/2/2003 3:27:50 AM Microsoft Corporation c:\winnt\system32\spoolss.dll
 spoolsv.exe 5.00.2195.4299 44.27 KB (45,328 bytes) 4/2/2003 3:27:50 AM Microsoft Corporation c:\winnt\system32\spoolsv.exe

clbcatq.dll	2000.2.3497.0	497.77 KB (509,712 bytes)	4/2/2003
1:16:26 PM	Microsoft Corporation	c:\winnt\system32\clbcatq.dll	
rpcss.dll	5.00.2195.5429	231.27 KB (236,816 bytes)	4/2/2003
1:16:42 PM	Microsoft Corporation	c:\winnt\system32\rpcss.dll	
svchost.exe	5.00.2134.1 7.77 KB (7,952 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\svchost.exe	
rdpwsx.dll	5.00.2195.5243	97.90 KB (100,248 bytes)	4/2/2003
1:16:42 PM	Microsoft Corporation	c:\winnt\system32\rdpwsx.dll	
ntlsapi.dll	5.00.2195.4907	6.77 KB (6,928 bytes)	
	12/31/1979 4:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\ntlsapi.dll		
mstlsapi.dll	5.00.2195.3895	25.77 KB (26,384 bytes)	4/2/2003
1:16:37 PM	Microsoft Corporation	c:\winnt\system32\mstlsapi.dll	
icaapi.dll	5.00.2195.3895	122.77 KB (125,712 bytes)	4/2/2003
1:16:31 PM	Microsoft Corporation	c:\winnt\system32\icaapi.dll	
regapi.dll	5.00.2195.5201	35.27 KB (36,112 bytes)	4/2/2003
1:16:42 PM	Microsoft Corporation	c:\winnt\system32\regapi.dll	
termsrv.exe	5.00.2195.5276	138.77 KB (142,096 bytes)	4/2/2003
1:16:45 PM	Microsoft Corporation	c:\winnt\system32\termsrv.exe	
dssenh.dll	5.00.2195.3665	142.77 KB (146,192 bytes)	4/2/2003
1:16:57 PM	Microsoft Corporation	c:\winnt\system32\dssenh.dll	
oakley.dll	5.00.2195.5326	382.27 KB (391,440 bytes)	4/2/2003
1:16:40 PM	Microsoft Corporation	c:\winnt\system32\oakley.dll	
mfc42u.dll	6.00.8665.0 972.05 KB (995,384 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\mfc42u.dll	
polagent.dll	5.00.2195.5428	94.77 KB (97,040 bytes)	4/2/2003
1:16:41 PM	Microsoft Corporation	c:\winnt\system32\polagent.dll	
scecli.dll	5.00.2195.4874	109.27 KB (111,888 bytes)	4/2/2003
1:16:43 PM	Microsoft Corporation	c:\winnt\system32\scecli.dll	
atl.dll	3.00.9435 73.06 KB (74,810 bytes)	4/2/2003 1:16:24 PM	
	Microsoft Corporation	c:\winnt\system32\atl.dll	
certcli.dll	5.00.2195.3649	130.27 KB (133,392 bytes)	4/2/2003
1:16:26 PM	Microsoft Corporation	c:\winnt\system32\certcli.dll	
esent.dll	6.0.3940.25 1.09 MB (1,137,936 bytes)	4/2/2003 1:16:29 PM	
	Microsoft Corporation	c:\winnt\system32\esent.dll	
ntdsatq.dll	5.00.2195.5246	31.27 KB (32,016 bytes)	4/2/2003
1:16:39 PM	Microsoft Corporation	c:\winnt\system32\ntdsatq.dll	
ntdsas.dll	5.00.2195.5438	1002.27 KB (1,026,320 bytes)	
	4/2/2003 1:16:39 PM	Microsoft Corporation	
	c:\winnt\system32\ntdsas.dll		
kdcsvc.dll	5.00.2195.5246	141.77 KB (145,168 bytes)	4/2/2003
1:16:33 PM	Microsoft Corporation	c:\winnt\system32\kdcsvc.dll	
sfmapi.dll	5.00.2134.1 38.77 KB (39,696 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\sfmapi.dll	
rassfm.dll	5.00.2195.4874	21.27 KB (21,776 bytes)	4/2/2003
1:16:42 PM	Microsoft Corporation	c:\winnt\system32\rassfm.dll	
rsabase.dll	5.00.2195.3839	128.27 KB (131,344 bytes)	
	7/22/2002 1:05:04 PM	Microsoft Corporation	
	c:\winnt\system32\rsabase.dll		
schannel.dll	5.00.2195.5284	139.27 KB (142,608 bytes)	
	12/31/1979 4:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\schannel.dll		

netlogon.dll	5.00.2195.5400	362.77 KB (371,472 bytes)	4/2/2003
1:16:33 PM	Microsoft Corporation	c:\winnt\system32\netlogon.dll	
kerberos.dll	5.00.2195.5201	20.27 KB (20,800 bytes)	4/2/2003
1:16:33 PM	Microsoft Corporation	c:\winnt\system32\kerberos.dll	
msprvs.dll	5.00.2154.1 41.50 KB (42,496 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\msprvs.dll	
samsrv.dll	5.00.2195.5201	374.27 KB (383,248 bytes)	
	12/31/1979 4:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\samsrv.dll		
lsasrv.dll	5.00.2195.5430	500.27 KB (512,272 bytes)	
	12/31/1979 4:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\lsasrv.dll		
lsass.exe	5.00.2195.5430	32.77 KB (33,552 bytes)	
	12/31/1979 4:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\lsass.exe		
xactsrv.dll	5.00.2134.1 90.27 KB (92,432 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\xactsrv.dll	
wmicore.dll	5.00.2195.3649	72.27 KB (74,000 bytes)	4/2/2003
1:16:47 PM	Microsoft Corporation	c:\winnt\system32\wmicore.dll	
rasadhlp.dll	5.00.2168.1 7.27 KB (7,440 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\rasadhlp.dll	
winnr.dll	5.00.2160.1 18.77 KB (19,216 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\winnr.dll	
mr20.dll	5.00.2195.4874	35.77 KB (36,624 bytes)	4/2/2003
1:16:42 PM	Microsoft Corporation	c:\winnt\system32\mr20.dll	
mswsock.dll	5.00.2195.4874	62.77 KB (64,272 bytes)	4/2/2003
1:16:38 PM	Microsoft Corporation	c:\winnt\system32\mswsock.dll	
msgsvcs.dll	5.00.2195.4874	34.77 KB (35,600 bytes)	
	12/31/1979 4:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\msgsvcs.dll		
browser.dll	5.00.2195.4874	48.77 KB (49,936 bytes)	4/2/2003
1:16:25 PM	Microsoft Corporation	c:\winnt\system32\browser.dll	
alrsvc.dll	5.00.2134.1 17.77 KB (18,192 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\alrsvc.dll	
trkwks.dll	5.00.2195.4874	88.77 KB (90,896 bytes)	4/2/2003
1:16:45 PM	Microsoft Corporation	c:\winnt\system32\trkwks.dll	
seclogon.dll	5.00.2195.5201	17.27 KB (17,680 bytes)	4/2/2003
1:16:43 PM	Microsoft Corporation	c:\winnt\system32\seclogon.dll	
psbase.dll	5.00.2195.4822	111.77 KB (114,448 bytes)	4/2/2003
1:16:41 PM	Microsoft Corporation	c:\winnt\system32\psbase.dll	
cryptsvc.dll	5.00.2195.4368	73.27 KB (75,024 bytes)	4/2/2003
1:16:27 PM	Microsoft Corporation	c:\winnt\system32\cryptsvc.dll	
cryptdll.dll	5.00.2135.1 41.27 KB (42,256 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\cryptdll.dll	
wkssvc.dll	5.00.2195.4874	95.27 KB (97,552 bytes)	
	12/31/1979 4:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\wkssvc.dll		
srvsvc.dll	5.00.2195.5400	81.77 KB (83,728 bytes)	
	12/31/1979 4:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\srvsvc.dll		
cfgmgr32.dll	5.00.2134.1 16.77 KB (17,168 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\cfgmgr32.dll	
dmserver.dll	2195.3649.297.3	12.27 KB (12,560 bytes)	4/2/2003
1:16:28 PM	VERITAS Software Corp.	c:\winnt\system32\dmserver.dll	

wshtcpip.dll	5.00.2195.4874	17.27 KB (17,680 bytes)	4/2/2003
1:16:34 PM	Microsoft Corporation	c:\winnt\system32\wshtcpip.dll	
lmhsvc.dll	5.00.2195.4874	9.77 KB (10,000 bytes)	
	12/31/1979 4:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\lmhsvc.dll		
dnssrslvr.dll	5.00.2195.5354	89.77 KB (91,920 bytes)	4/2/2003
1:16:28 PM	Microsoft Corporation	c:\winnt\system32\dnssrslvr.dll	
tapi32.dll	5.00.2182.1 123.27 KB (126,224 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\tapi32.dll	
rasman.dll	5.00.2195.5292	54.77 KB (56,080 bytes)	
	12/31/1979 4:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\rasman.dll		
rasapi32.dll	5.00.2195.5438	191.77 KB (196,368 bytes)	
	12/31/1979 4:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\rasapi32.dll		
rtutils.dll	5.00.2168.1 43.77 KB (44,816 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\rtutils.dll	
adslsdp.dll	5.00.2195.5400	127.77 KB (130,832 bytes)	4/2/2003
1:16:24 PM	Microsoft Corporation	c:\winnt\system32\adslsdp.dll	
activeds.dll	5.00.2195.5312	175.27 KB (179,472 bytes)	4/2/2003
1:16:22 PM	Microsoft Corporation	c:\winnt\system32\activeds.dll	
oleaut32.dll	2.40.4518 612.27 KB (626,960 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\oleaut32.dll	
mprapi.dll	5.00.2181.1 79.27 KB (81,168 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\mprapi.dll	
iphlpapi.dll	5.00.2195.2 68.27 KB (69,904 bytes)	4/2/2003 1:16:32 PM	
	Microsoft Corporation	c:\winnt\system32\iphlpapi.dll	
icmp.dll	5.00.2134.1 7.27 KB (7,440 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\icmp.dll	
dhcpcsvc.dll	5.00.2195.4874	87.77 KB (89,872 bytes)	
	12/31/1979 4:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\dhcpcsvc.dll		
eventlog.dll	5.00.2195.5336	44.27 KB (45,328 bytes)	4/2/2003
1:16:30 PM	Microsoft Corporation	c:\winnt\system32\eventlog.dll	
ntdsapi.dll	5.00.2195.4827	56.27 KB (57,616 bytes)	4/2/2003
1:16:39 PM	Microsoft Corporation	c:\winnt\system32\ntdsapi.dll	
scsersv.dll	5.00.2195.5316	242.77 KB (248,592 bytes)	4/2/2003
1:16:43 PM	Microsoft Corporation	c:\winnt\system32\scsersv.dll	
umpnprmgr.dll	5.00.2182.1	86.27 KB (88,336 bytes)	
	12/31/1979 4:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\umpnprmgr.dll		
services.exe	5.00.2195.3940	86.77 KB (88,848 bytes)	
	12/31/1979 4:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\services.exe		
msv1_0.dll	5.00.2195.4745	112.27 KB (114,960 bytes)	
	12/31/1979 4:00:00 PM	Microsoft Corporation	
	c:\winnt\system32\msv1_0.dll		
lz32.dll	5.00.2134.1 9.77 KB (10,000 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\lz32.dll	
version.dll	5.00.2134.1 15.77 KB (16,144 bytes)	12/31/1979 4:00:00	
PM	Microsoft Corporation	c:\winnt\system32\version.dll	

winspool.drv	5.00.2195.5225	111.27 KB (113,936 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\winspool.drv				
winscard.dll	5.00.2134.1	77.27 KB (79,120 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\winscard.dll				
wlnotify.dll	5.00.2195.5377	54.27 KB (55,568 bytes)	4/2/2003 1:16:47 PM	Microsoft Corporation
c:\winnt\system32\wlnotify.dll				
cscdll.dll	5.00.2195.5434	98.77 KB (101,136 bytes)	4/2/2003 1:16:27 PM	Microsoft Corporation
c:\winnt\system32\cscdll.dll				
rsaenh.dll	5.00.2195.3839	130.77 KB (133,904 bytes)	4/2/2003 1:16:58 PM	Microsoft Corporation
c:\winnt\system32\rsaenh.dll				
mscat32.dll	5.131.2134.1	7.77 KB (7,952 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\mscat32.dll				
ole32.dll	5.00.2195.5400	968.27 KB (991,504 bytes)	4/2/2003 1:16:40 PM	Microsoft Corporation
c:\winnt\system32\ole32.dll				
imagehlp.dll	5.00.2195.5242	125.77 KB (128,784 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\imagehlp.dll				
msasn1.dll	5.00.2195.4067	51.27 KB (52,496 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\msasn1.dll				
crypt32.dll	5.131.2195.4558	464.27 KB (475,408 bytes)	4/2/2003 1:16:27 PM	Microsoft Corporation
c:\winnt\system32\crypt32.dll				
wintrust.dll	5.131.2195.3775	162.27 KB (166,160 bytes)	4/2/2003 1:16:47 PM	Microsoft Corporation
c:\winnt\system32\wintrust.dll				
mpr.dll	5.00.2195.3649	53.77 KB (55,056 bytes)	4/2/2003 1:16:34 PM	Microsoft Corporation
c:\winnt\system32\mpr.dll				
shlwapi.dll	5.00.3502.5332	283.27 KB (290,064 bytes)	4/2/2003 1:16:44 PM	Microsoft Corporation
c:\winnt\system32\shlwapi.dll				
shell32.dll	5.00.3502.5436	2.26 MB (2,374,416 bytes)	4/2/2003 1:16:43 PM	Microsoft Corporation
c:\winnt\system32\shell32.dll				
msgina.dll	5.00.2195.4733	324.77 KB (332,560 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\msgina.dll				
comctl32.dll	5.81	539.27 KB (552,208 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\comctl32.dll				
setupapi.dll	5.00.2195.5400	553.77 KB (567,056 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\setupapi.dll				
winmm.dll	5.00.2161.1	184.77 KB (189,200 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\winmm.dll				
winsta.dll	5.00.2195.4655	36.77 KB (37,648 bytes)	4/2/2003 1:16:47 PM	Microsoft Corporation
c:\winnt\system32\winsta.dll				
wssock32.dll	5.00.2195.4874	21.27 KB (21,776 bytes)	4/2/2003 1:16:47 PM	Microsoft Corporation
c:\winnt\system32\wssock32.dll				
dnsapi.dll	5.00.2195.5354	131.27 KB (134,416 bytes)	4/2/2003 1:16:28 PM	Microsoft Corporation
c:\winnt\system32\dnsapi.dll				
wldap32.dll	5.00.2195.5400	158.77 KB (162,576 bytes)	4/2/2003 1:16:47 PM	Microsoft Corporation
c:\winnt\system32\wldap32.dll				
ws2help.dll	5.00.2134.1	17.77 KB (18,192 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\ws2help.dll				

ws2_32.dll	5.00.2195.4874	66.77 KB (68,368 bytes)	4/2/2003 12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\ws2_32.dll				
netapi32.dll	5.00.2134.1	11.27 KB (11,536 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\netapi32.dll				
netrap.dll	5.00.2195.5427	305.27 KB (312,592 bytes)	4/2/2003 1:16:38 PM	Microsoft Corporation
c:\winnt\system32\netapi32.dll				
profmap.dll	5.00.2181.1	29.27 KB (29,968 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\profmap.dll				
secur32.dll	5.00.2195.4587	47.27 KB (48,400 bytes)	4/2/2003 1:16:43 PM	Microsoft Corporation
c:\winnt\system32\secur32.dll				
sfc.dll	5.00.2195.3649	92.11 KB (94,320 bytes)	4/2/2003 1:16:43 PM	Microsoft Corporation
c:\winnt\system32\sfc.dll				
nddeapi.dll	5.00.2195.4509	15.77 KB (16,144 bytes)	4/2/2003 1:16:38 PM	Microsoft Corporation
c:\winnt\system32\nddeapi.dll				
userenv.dll	5.00.2195.5425	363.77 KB (372,496 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\userenv.dll				
user32.dll	5.00.2195.4314	395.77 KB (405,264 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\user32.dll				
gdi32.dll	5.00.2195.5252	228.77 KB (234,256 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\gdi32.dll				
rpcrt4.dll	5.00.2195.5419	440.27 KB (450,832 bytes)	4/2/2003 1:16:42 PM	Microsoft Corporation
c:\winnt\system32\rpcrt4.dll				
advapi32.dll	5.00.2195.5385	358.77 KB (367,376 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\advapi32.dll				
kernel32.dll	5.00.2195.5400	716.77 KB (733,968 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\kernel32.dll				
msvcrt.dll	6.10.9359.0	284.05 KB (290,869 bytes)	7/22/2002 1:05:04 PM	Microsoft Corporation
c:\winnt\system32\msvcrt.dll				
winlogon.exe	5.00.2195.5386	174.77 KB (178,960 bytes)	4/2/2003 1:16:46 PM	Microsoft Corporation
c:\winnt\system32\winlogon.exe				
sfcfiles.dll	5.00.2195.5426	951.27 KB (974,096 bytes)	4/2/2003 1:16:43 PM	Microsoft Corporation
c:\winnt\system32\sfcfiles.dll				
ntdll.dll	5.00.2195.5400	479.27 KB (490,768 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\ntdll.dll				
smss.exe	5.00.2195.5382	44.77 KB (45,840 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\smss.exe				

[Services]

Display Name	Name	State	Start Mode	Service
Alert	Path	Error Control	Start Name	Tag ID
Alert	Alert	Running	Auto	Share Process
	c:\winnt\system32\services.exe			Normal
	LocalSystem			0

Application Management	AppMgmt	Stopped	Manual	Share
Process	c:\winnt\system32\services.exe		Normal	
Background Intelligent Transfer Service	BITS	Stopped	Manual	
Share Process	c:\winnt\system32\svchost.exe -k			
bitsgroup	Normal	LocalSystem	0	
Computer Browser	Browser	Running	Auto	Share
Process	c:\winnt\system32\services.exe		Normal	
Indexing Service	cisvc	Stopped	Manual	Share
Process	c:\winnt\system32\cisvc.exe		Normal	
ClipBook	ClipSrv	Stopped	Manual	Own Process
	c:\winnt\system32\clipsrv.exe		Normal	
Compaq Remote Monitor Service	CpqRcmC	Running	Auto	
Own Process	c:\winnt\system32\cpqrcmc.exe		Normal	
Distributed File System	Dfs	Running	Auto	Own
Process	c:\winnt\system32\dfssvc.exe		Normal	
DHCP Client	Dhcp	Running	Auto	Share Process
	c:\winnt\system32\services.exe		Normal	
Logical Disk Manager	Administrative Service			
Manual	Share Process			
	c:\winnt\system32\dmadmin.exe /com		Normal	
Logical Disk Manager	dmserver	Running	Auto	Share
Process	c:\winnt\system32\services.exe		Normal	
DNS Client	Dnscache	Running	Auto	Share Process
	c:\winnt\system32\services.exe		Normal	
Event Log	Eventlog	Running	Auto	Share Process
	c:\winnt\system32\services.exe		Normal	
COM+ Event System	EventSystem	Running	Manual	Share
Process	c:\winnt\system32\svchost.exe -k netsvcs		Normal	
Fax Service	Fax	Stopped	Manual	Own Process
	c:\winnt\system32\faxsvc.exe		Normal	
IIS Admin Service	IISADMIN	Running	Auto	Share
Process	c:\winnt\system32\inetrv\inetinfo.exe		Normal	
Intersite Messaging	IsmServ	Stopped	Disabled	Own
Process	c:\winnt\system32\ismerv.exe		Normal	
Kerberos Key Distribution Center	kdc	Stopped	Disabled	
Share Process	c:\winnt\system32\lsass.exe			
	Normal	LocalSystem	0	

Server	Ianmanserver	Running	Auto	Share Process
	c:\winnt\system32\services.exe			Normal
Workstation	Ianmanworkstation	Running	Auto	Share
Process	c:\winnt\system32\services.exe			Normal
License Logging Service	LicenseService	Running	Auto	
	Own Process c:\winnt\system32\lssrv.exe			Normal
TCP/IP NetBIOS Helper Service	LmHosts	Running	Auto	
	Share Process c:\winnt\system32\services.exe			Normal
Messenger	Messenger	Running	Auto	Share Process
	c:\winnt\system32\services.exe			Normal
NetMeeting Remote Desktop Sharing	mnmsrvc	Stopped	Manual	
	Own Process c:\winnt\system32\mnmsrvc.exe			Normal
Distributed Transaction Coordinator	MSDTC	Running	Auto	
	Own Process c:\winnt\system32\msdtc.exe			Normal
Windows Installer	MSIServer	Stopped	Manual	Share
Process	c:\winnt\system32\msiexec.exe /v			Normal
Network DDE	NetDDE	Stopped	Manual	Share Process
	c:\winnt\system32\netdde.exe			Normal
Network DDE DSDM	NetDDEdsdm	Stopped	Manual	Share
Process	c:\winnt\system32\netdde.exe			Normal
Net Logon	Netlogon	Stopped	Manual	Share Process
	c:\winnt\system32\lsass.exe			Normal
Network Connections	Netman	Running	Manual	Share
Process	c:\winnt\system32\svchost.exe -k netsvcs			Normal
File Replication	NtFrs	Stopped	Manual	Own
Process	c:\winnt\system32\ntfrs.exe			Ignore
NT LM Security Support Provider	NtLmSsp	Stopped	Manual	
	Share Process c:\winnt\system32\lsass.exe			Normal
Removable Storage	NtmsSvc	Running	Auto	Share
Process	c:\winnt\system32\svchost.exe -k netsvcs			Normal
Plug and Play	PlugPlay	Running	Auto	Share Process
	c:\winnt\system32\services.exe			Normal
IPSEC Policy Agent	PolicyAgent	Running	Auto	Share
Process	c:\winnt\system32\lsass.exe			Normal

Protected Storage	ProtectedStorage	Running	Auto	
	Share Process c:\winnt\system32\services.exe			Normal
Remote Access Connection Manager	RasMan	Stopped	Manual	
	Share Process c:\winnt\system32\svchost.exe -k netsvcs			Normal
Remote Access Connection Manager	RasMan	Stopped	Manual	
	Share Process c:\winnt\system32\svchost.exe -k netsvcs			Normal
Routing and Remote Access	RemoteAccess	Stopped		
	Disabled Share Process c:\winnt\system32\svchost.exe -k netsvcs			Normal
Remote Registry Service	RemoteRegistry	Running	Auto	
	Own Process c:\winnt\system32\regsvc.exe			Normal
Remote Command Service	RMSYS	Running	Auto	Own
Process	c:\program files\benchcraft\rsys.exe			Normal
Remote Procedure Call (RPC) Locator	RpcLocator	Stopped	Manual	
	Own Process c:\winnt\system32\locator.exe			Normal
Remote Procedure Call (RPC)	RpcSs	Running	Auto	
	Share Process c:\winnt\system32\svchost -k rpcss			Normal
QoS RSVP	RSVP	Running	Manual	Own Process
	c:\winnt\system32\rsvp.exe -s			Normal
Security Accounts Manager	SamSs	Running	Auto	Share
Process	c:\winnt\system32\lsass.exe			Normal
Smart Card Helper	SCardDrv	Stopped	Manual	Share
Process	c:\winnt\system32\scardsvr.exe			Ignore
Smart Card	SCardSvr	Stopped	Manual	Share Process
	c:\winnt\system32\scardsvr.exe			Ignore
Task Scheduler	Schedule	Running	Auto	Share
Process	c:\winnt\system32\mstask.exe			Normal
RunAs Service	seclogon	Running	Auto	Share
Process	c:\winnt\system32\services.exe			Ignore
System Event Notification	SENS	Running	Auto	Share
Process	c:\winnt\system32\svchost.exe -k netsvcs			Normal
Internet Connection Sharing	SharedAccess	Stopped		
	Manual Share Process c:\winnt\system32\svchost.exe -k netsvcs			Normal

Simple Mail Transport Protocol (SMTP)	SMTPSVC	Running	Auto	
	Share Process c:\winnt\system32\inetinfo.exe			Normal
SNMP Service	SNMP	Running	Auto	Own
Process	c:\winnt\system32\snmp.exe			Normal
SNMP Trap Service	SNMPTRAP	Stopped	Manual	Own
Process	c:\winnt\system32\snmptrap.exe			Normal
Print Spooler	Spooler	Running	Auto	Own Process
	c:\winnt\system32\spoolsv.exe			Normal
hp ProLiant System Shutdown Service	sysdown	Running	Auto	
	Own Process c:\winnt\system32\sysdown.exe			Normal
Performance Logs and Alerts	SysmonLog	Stopped	Manual	
	Own Process c:\winnt\system32\smlogsvc.exe			Normal
Telephony	TapiSrv	Running	Manual	Share Process
	c:\winnt\system32\svchost.exe -k tapisrv			Normal
Terminal Services	TermService	Running	Auto	Own
Process	c:\winnt\system32\termsrv.exe			Normal
Telnet	TlntSvr	Stopped	Manual	Own Process
	c:\winnt\system32\tlntsvr.exe			Normal
Distributed Link Tracking Server	TrkSvr	Stopped	Manual	
	Share Process c:\winnt\system32\services.exe			Normal
Distributed Link Tracking Client	TrkWks	Running	Auto	
	Share Process c:\winnt\system32\services.exe			Normal
Uninterruptible Power Supply	UPS	Stopped	Manual	
	Own Process c:\winnt\system32\ups.exe			Normal
Utility Manager	UtilMan	Stopped	Manual	Own
Process	c:\winnt\system32\utilman.exe			Normal
Windows Time	W32Time	Stopped	Manual	Share
Process	c:\winnt\system32\services.exe			Normal
World Wide Web Publishing Service	W3SVC	Running	Auto	
	Share Process c:\winnt\system32\inetinfo.exe			Normal
Windows Management Instrumentation	WinMgmt	Running	Auto	
	Own Process c:\winnt\system32\wbem\winmgmt.exe			Ignore
Windows Management Instrumentation Driver Extensions				Wmi
	Running Manual Share Process c:\winnt\system32\services.exe			Normal

Automatic Updates wuauerv Running Auto Share
 Process c:\winnt\system32\svchost.exe -k wugroup Normal
 LocalSystem 0

[Program Groups]

Group Name	Name	User Name
Accessories	Default User:Accessories	Default User
Accessories\Accessibility	Default User:Accessories\Accessibility	Default User
Accessories\Entertainment	Default User:Accessories\Entertainment	Default User
Accessories\System Tools	Default User:Accessories\System Tools	Default User
Startup	Default User:Startup	Default User
Accessories	All Users:Accessories	All Users
Accessories\Accessibility	All Users:Accessories\Accessibility	All Users
Accessories\Communications	All Users:Accessories\Communications	All Users
Accessories\Entertainment	All Users:Accessories\Entertainment	All Users
Accessories\Games	All Users:Accessories\Games	All Users
Accessories\Microsoft Script Debugger	All Users:Accessories\Microsoft Script Debugger	All Users
Accessories\System Tools	All Users:Accessories\System Tools	All Users
Administrative Tools	All Users:Administrative Tools	All Users
Compaq System Tools	All Users:Compaq System Tools	All Users
Microsoft SQL Server	All Users:Microsoft SQL Server	All Users
Startup	All Users:Startup	All Users
Accessories	W11\Administrator:Accessories	W11\Administrator
Accessories\Accessibility	W11\Administrator:Accessories\Accessibility	W11\Administrator
Accessories\Entertainment	W11\Administrator:Accessories\Entertainment	W11\Administrator
Accessories\System Tools	W11\Administrator:Accessories\System Tools	W11\Administrator
Administrative Tools	W11\Administrator:Administrative Tools	W11\Administrator
Benchcraft	W11\Administrator:Benchcraft	W11\Administrator
QLogic Corporation	W11\Administrator:QLogic Corporation	W11\Administrator
QLogic Corporation\SANblade Control VIX	W11\Administrator:QLogic Corporation\SANblade Control VIX	W11\Administrator
Startup	W11\Administrator:Startup	W11\Administrator

[Startup Programs]

Program	Command	User Name	Location
CPQTEAM	cpqteam.exe	All Users	HKL\M\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

[OLE Registration]

Object	Local Server
Sound (OLE2)	sndrec32.exe
Media Clip	mplay32.exe
Video Clip	mplay32.exe /avi
MIDI Sequence	mplay32.exe /mid
Sound	Not Available
Media Clip	Not Available
Image Document	"C:\Program Files\Windows NT\Accessories\ImageVue\KodakImg.exe"
WordPad Document	"%ProgramFiles%\Windows NT\Accessories\WORDPAD.EXE"
Windows Media Services DRM Storage object	Not Available
Bitmap Image	mspaint.exe

[Internet Explorer 5]

[Following are sub-categories of this main category]

[Summary]

Item	Value
Version	5.00.3502.1000
Build	53502.1000
Product ID	51876-270-2828816-05612
Application Path	C:\Program Files\Internet Explorer
Language	English (United States)
Active Printer	Not Available

Cipher Strength	168-bit
Content Advisor	Disabled
IEAK Install	No

[File Versions]

File	Version	Size	Date	Path	Company
advapi32.dll	5.0.2195.5385	359 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
advpack.dll	5.0.3502.4373	87 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
browsecl.dll	5.0.3502.4373	35 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
browserui.dll	5.0.3502.4373	791 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
ckcnv.exe	5.0.2189.1	9 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
comctl32.dll	5.81.3315.3727	539 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
crypt32.dll	5.131.2195.4558	464 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
enhsg.dll	<File Missing>		Not Available	Not Available	Not Available
iemigrat.dll	<File Missing>		Not Available	Not Available	Not Available

iesetup.dll	5.0.3502.4373	57 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
imagehlp.dll	5.0.2195.5242	126 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
imghelp.dll	<File Missing>		Not Available	Not Available	Not Available
inseng.dll	5.0.3502.4373	72 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
jobexec.dll	5.0.0.1	47 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
jscrip.dll	5.1.0.5907	476 KB	7/22/2002 12:05:04 PM	C:\WINNT\system32	Microsoft Corporation
jsproxy.dll	5.0.2920.0	13 KB	12/7/1999 4:00:00 AM	C:\WINNT\system32	Microsoft Corporation
msahtml.dll	<File Missing>		Not Available	Not Available	Not Available
mshtml.dll	5.0.3502.5390	2284 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
msjava.dll	5.0.3805.0	924 KB	7/22/2002 12:05:04 PM	C:\WINNT\system32	Microsoft Corporation
msoss.dll	<File Missing>		Not Available	Not Available	Not Available
msxml.dll	8.0.6730.0	494 KB	7/22/2002 12:05:04 PM	C:\WINNT\system32	Microsoft Corporation
occache.dll	5.0.3315.3727	86 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
ole32.dll	5.0.2195.5400	968 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
oleaut32.dll	5.0.4518.0	612 KB	7/22/2002 12:05:04 PM	C:\WINNT\system32	Microsoft Corporation
olepro32.dll	5.0.4518.0	160 KB	7/22/2002 12:05:04 PM	C:\WINNT\system32	Microsoft Corporation
rsabase.dll	5.0.2195.3839	128 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
rsaenh.dll	5.0.2195.3839	131 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
rsapi32.dll	<File Missing>		Not Available	Not Available	Not Available
rsasig.dll	<File Missing>		Not Available	Not Available	Not Available
schannel.dll	5.1.2195.0	139 KB	7/22/2002 12:05:04 PM	C:\WINNT\system32	Microsoft Corporation
shdoc401.dll	<File Missing>		Not Available	Not Available	Not Available
shdocvw.dll	5.0.3502.5400	1079 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
shell32.dll	5.0.3502.5436	2319 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
shlwapi.dll	5.0.3502.5332	283 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation
url.dll	5.0.3502.4510	82 KB	7/22/2002 12:05:04	C:\WINNT\system32	Microsoft Corporation

urlmon.dll	5.0.3502.5400	442 KB	7/22/2002 12:05:04 PM
PM	C:\WINNT\system32	Microsoft Corporation	7/22/2002 12:05:04 PM
vbscript.dll	5.1.0.7426	428 KB	
	C:\WINNT\system32	Microsoft Corporation	
webcheck.dll	5.0.3315.3727	251 KB	7/22/2002 12:05:04 PM
PM	C:\WINNT\system32	Microsoft Corporation	
win.com	5.0.2134.1	24 KB	12/7/1999 4:00:00 AM
	C:\WINNT\system32	Microsoft Corporation	
wininet.dll	5.0.3502.4619	451 KB	7/22/2002 12:05:04 PM
PM	C:\WINNT\system32	Microsoft Corporation	
winsock.dll	3.10.0.103	3 KB	12/7/1999 4:00:00 AM
	C:\WINNT\system32	Microsoft Corporation	
wintrust.dll	5.131.2195.3775	162 KB	7/22/2002 12:05:04 PM
PM	C:\WINNT\system32	Microsoft Corporation	
wsock.vxd	<File Missing>	Not Available	Not Available
Available	Not Available		
wsock32.dll	5.0.2195.4874	21 KB	7/22/2002 12:05:04 PM
PM	C:\WINNT\system32	Microsoft Corporation	
wsock32n.dll	<File Missing>	Not Available	Not Available
Available	Not Available		

[Connectivity]

Item	Value
Connection Preference	Never dial
EnableHttp1.1	1
ProxyHttp1.1 0	

LAN Settings

AutoConfigProxy	wininet.dll
AutoProxyDetectMode	Disabled
AutoConfigURL	
Proxy	Disabled
ProxyServer	
ProxyOverride	

[Cache]

[Following are sub-categories of this main category]

[Summary]

Item	Value
Page Refresh Type	Automatic
Temporary Internet Files Folder	C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files
Total Disk Space	34719 MB
Available Disk Space	31724 MB
Maximum Cache Size	1084 MB
Available Cache Size	1085 MB

[List of Objects]
Program File Status CodeBase
No cached object information available

[Content]

[Following are sub-categories of this main category]

[Summary]

Item	Value
Content Advisor	Disabled

[Personal Certificates]

Issued To	Issued By	Validity	Signature Algorithm
Administrator	Administrator	4/2/2003 to 3/9/2103	sha1RSA

[Other People Certificates]

Issued To	Issued By	Validity	Signature Algorithm
No other people certificate information available			

[Publishers]

Name
No publisher information available

[Security]

Zone	Security Level
Local intranet	Medium-low
Trusted sites	Low
Internet	Medium
Restricted sites	High

C.3 RTE Input Parameters

3Tier.pro

Profile: 63Kwh_32W_124DE_balanced
File Path: C:\Benchcraft\63Kwh_32W_124DE_balanced.pro
Version: 4

Number of Engines: 124

Name: DRIVER001
Description: Driver c3.1
Directory: c:\3.1.log
Machine: c3.hp-perf.net
Parameter Set: Optimized
Index: 0
Seed: 69483
Configured Users: 5200
Pipe Name: DRIVER13644875
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp0

Name: DRIVER002
Description: Driver c3.2
Directory: c:\3.2.log
Machine: c3.hp-perf.net
Parameter Set: Optimized
Index: 10000000
Seed: 69483
Configured Users: 5200
Pipe Name: DRIVER2979718
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp0

Name: DRIVER003
Description: Driver c4.1
Directory: c:\4.1.log
Machine: c4.hp-perf.net
Parameter Set: Optimized
Index: 20000000
Seed: 69483
Configured Users: 5200
Pipe Name: DRIVER3854634281
Connect Rate: 500
Start Rate: 150

Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp0

Name: DRIVER004
Description: Driver cl4.2
Directory: c:\4.2.log
Machine: cl4.hp-perf.net
Parameter Set: Optimized
Index: 30000000
Seed: 69483
Configured Users: 5200
Pipe Name: DRIVER4854658406
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp1

Name: DRIVER005
Description: Driver cl5.1
Directory: c:\5.1.log
Machine: cl5.hp-perf.net
Parameter Set: Optimized
Index: 40000000
Seed: 69483
Configured Users: 5200
Pipe Name: DRIVER5854679359
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp1

Name: DRIVER006
Description: Driver cl5.2
Directory: c:\5.2.log
Machine: cl5.hp-perf.net
Parameter Set: Optimized
Index: 50000000
Seed: 69483
Configured Users: 5200
Pipe Name: DRIVER6854746406
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1

CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp1

Name: DRIVER007
Description: Driver cl6.1
Directory: c:\6.1.log
Machine: cl6.hp-perf.net
Parameter Set: Optimized
Index: 60000000
Seed: 69483
Configured Users: 4950
Pipe Name: DRIVER7854769640
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp1

Name: DRIVER008
Description: Driver cl6.2
Directory: c:\6.2.log
Machine: cl6.hp-perf.net
Parameter Set: Optimized
Index: 70000000
Seed: 69483
Configured Users: 4950
Pipe Name: DRIVER8854798000
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp2

Name: DRIVER009
Description: Driver cl7.1
Directory: c:\7.1.log
Machine: cl7.hp-perf.net
Parameter Set: Optimized
Index: 80000000
Seed: 69483
Configured Users: 4950
Pipe Name: DRIVER9854821718
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0

Additional Options: spprefix=sp2
Name: DRIVER010
Description: Driver cl7.2
Directory: c:\7.2.log
Machine: cl7.hp-perf.net
Parameter Set: Optimized
Index: 90000000
Seed: 69483
Configured Users: 4950
Pipe Name: DRIVER10854846593
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp2

Name: DRIVER011
Description: Driver cl8.1
Directory: c:\8.1.log
Machine: cl8.hp-perf.net
Parameter Set: Optimized
Index: 100000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER11854931343
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp2

Name: DRIVER012
Description: Driver cl8.2
Directory: c:\8.2.log
Machine: cl8.hp-perf.net
Parameter Set: Optimized
Index: 110000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER12855001093
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp3

Name: DRIVER013

Description: Driver c9.1
 Directory: c:\9.1.log
 Machine: c9.hp-perf.net
 Parameter Set: Optimized
 Index: 120000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER13855032656
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options: spprefix=sp3

Name: DRIVER014
 Description: Driver c9.2
 Directory: c:\9.2.log
 Machine: c9.hp-perf.net
 Parameter Set: Optimized
 Index: 130000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER14855053250
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options: spprefix=sp3

Name: DRIVER015
 Description: Driver d10.1
 Directory: c:\10.1.log
 Machine: c10.hp-perf.net
 Parameter Set: Optimized
 Index: 140000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER15855082328
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options: spprefix=sp3

Name: DRIVER016
 Description: Driver d10.2

Name: DRIVER017
 Description: Driver c11.1
 Directory: c:\11.1.log
 Machine: c11.hp-perf.net
 Parameter Set: Optimized
 Index: 160000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER17855188843
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options: spprefix=sp4

Name: DRIVER018
 Description: Driver c11.2
 Directory: c:\11.2.log
 Machine: c11.hp-perf.net
 Parameter Set: Optimized
 Index: 170000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER18855324406
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options: spprefix=sp4

Name: DRIVER019
 Description: Driver c12.1
 Directory: c:\12.1.log
 Machine: c12.hp-perf.net
 Parameter Set: Optimized

Name: DRIVER020
 Description: Driver c12.2
 Directory: c:\12.2.log
 Machine: c12.hp-perf.net
 Parameter Set: Optimized

Name: DRIVER021
 Description: Driver c13.1
 Directory: c:\13.1.log
 Machine: c13.hp-perf.net
 Parameter Set: Optimized
 Index: 200000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER20855403031
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options: spprefix=sp5

Name: DRIVER022
 Description: Driver c13.2
 Directory: c:\13.2.log
 Machine: c13.hp-perf.net
 Parameter Set: Optimized
 Index: 210000000
 Seed: 69483
 Configured Users: 5130

Name: DRIVER023
 Description: Driver c13.3
 Directory: c:\13.3.log
 Machine: c13.hp-perf.net
 Parameter Set: Optimized
 Index: 220000000
 Seed: 69483
 Configured Users: 5130

Name: DRIVER024
 Description: Driver c13.4
 Directory: c:\13.4.log
 Machine: c13.hp-perf.net
 Parameter Set: Optimized
 Index: 230000000
 Seed: 69483
 Configured Users: 5130

Pipe Name: DRIVER22855439000
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp5

Name: DRIVER023
Description: Driver cl14.1
Directory: c:\14.1.log
Machine: cl14.hp-perf.net
Parameter Set: Optimized
Index: 220000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER23855574843
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp5

Name: DRIVER024
Description: Driver cl14.2
Directory: c:\14.2.log
Machine: cl14.hp-perf.net
Parameter Set: Optimized
Index: 230000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER24855750390
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp6

Name: DRIVER025
Description: Driver cl15.1
Directory: c:\15.1.log
Machine: cl15.hp-perf.net
Parameter Set: Optimized
Index: 240000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER25855777000

Start Rate: 150
Connect Rate: 500
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp6

Name: DRIVER026
Description: Driver cl15.2
Directory: c:\15.2.log
Machine: cl15.hp-perf.net
Parameter Set: Optimized
Index: 250000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER26855805562
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp6

Name: DRIVER027
Description: Driver cl16.1
Directory: c:\16.1.log
Machine: cl16.hp-perf.net
Parameter Set: Optimized
Index: 260000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER27855831718
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp6

Name: DRIVER028
Description: Driver cl16.2
Directory: c:\16.2.log
Machine: cl16.hp-perf.net
Parameter Set: Optimized
Index: 270000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER28855860406
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1

CLIENT_NURAND: 233
Connect Rate: 500
CPU: 1
Additional Options: spprefix=sp7

Name: DRIVER029
Description: Driver cl17.1
Directory: c:\17.1.log
Machine: cl17.hp-perf.net
Parameter Set: Optimized
Index: 280000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER29855926250
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp7

Name: DRIVER030
Description: Driver cl17.2
Directory: c:\17.2.log
Machine: cl17.hp-perf.net
Parameter Set: Optimized
Index: 290000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER30855949312
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp7

Name: DRIVER031
Description: Driver cl18.1
Directory: c:\18.1.log
Machine: cl18.hp-perf.net
Parameter Set: Optimized
Index: 300000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER31855971765
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0

Additional Options: spprefix=sp7

Name: DRIVER032
Description: Driver cl18.2
Directory: c:\18.2.log
Machine: cl18.hp-perf.net
Parameter Set: Optimized
Index: 310000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER32600618984
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER033
Description: Driver cl19.1
Directory: c:\19.1.log
Machine: cl19.hp-perf.net
Parameter Set: Optimized
Index: 320000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER33602687968
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER034
Description: Driver cl19.2
Directory: c:\19.2.log
Machine: cl19.hp-perf.net
Parameter Set: Optimized
Index: 330000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER34602718093
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER035
Description: Driver cl20.1
Directory: c:\20.1.log
Machine: cl20.hp-perf.net
Parameter Set: Optimized
Index: 340000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER35602756921
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER036
Description: Driver cl20.2
Directory: c:\20.2.log
Machine: cl20.hp-perf.net
Parameter Set: Optimized
Index: 350000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER36602790046
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER037
Description: Driver cl21.1
Directory: c:\21.1.log
Machine: cl21.hp-perf.net
Parameter Set: Optimized
Index: 360000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER37602990203
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER038
Description: Driver cl21.2

Name: DRIVER039
Description: Driver cl21.2
Directory: c:\21.2.log
Machine: cl21.hp-perf.net
Parameter Set: Optimized
Index: 370000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER38603018968
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER039
Description: Driver cl22.1
Directory: c:\22.1.log
Machine: cl22.hp-perf.net
Parameter Set: Optimized
Index: 380000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER39603064187
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER040
Description: Driver cl22.1
Directory: c:\22.2.log
Machine: cl22.hp-perf.net
Parameter Set: Optimized
Index: 390000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER40603090500
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER041
Description: Driver cl23.1
Directory: c:\23.1.log
Machine: cl23.hp-perf.net
Parameter Set: Optimized

Index: 400000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER41603731140
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER042
Description: Driver cl23.2
Directory: c:\23.2.log
Machine: cl23.hp-perf.net
Parameter Set: Optimized
Index: 410000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER42603751468
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER043
Description: Driver cl24.1
Directory: c:\24.1.log
Machine: cl24.hp-perf.net
Parameter Set: Optimized
Index: 420000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER43603851921
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER044
Description: Driver cl24.2
Directory: c:\24.2.log
Machine: cl24.hp-perf.net
Parameter Set: Optimized
Index: 430000000

Configured Users: 5130
Pipe Name: DRIVER44603871796
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER045
Description: Driver cl25.1
Directory: c:\25.1.log
Machine: cl25.hp-perf.net
Parameter Set: Optimized
Index: 440000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER45604247984
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER046
Description: Driver cl25.2
Directory: c:\25.2.log
Machine: cl25.hp-perf.net
Parameter Set: Optimized
Index: 450000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER46604281703
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER047
Description: Driver cl26.1
Directory: c:\26.1.log
Machine: cl26.hp-perf.net
Parameter Set: Optimized
Index: 460000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER47604355625

Start Rate: 150
Connect Rate: 500
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER048
Description: Driver cl26.2
Directory: c:\26.2.log
Machine: cl26.hp-perf.net
Parameter Set: Optimized
Index: 470000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER48604382093
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER049
Description: Driver cl27.1
Directory: c:\27.1.log
Machine: cl27.hp-perf.net
Parameter Set: Optimized
Index: 480000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER49604582171
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER050
Description: Driver cl27.2
Directory: c:\27.2.log
Machine: cl27.hp-perf.net
Parameter Set: Optimized
Index: 490000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER50604653093
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1

Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

 Name: DRIVER051
 Description: Driver cl28.1
 Directory: c:\28.1.log
 Machine: cl28.hp-perf.net
 Parameter Set: Optimized
 Index: 500000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER51604674125
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

 Name: DRIVER052
 Description: Driver cl28.2
 Directory: c:\28.2.log
 Machine: cl28.hp-perf.net
 Parameter Set: Optimized
 Index: 510000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER52604703187
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

 Name: DRIVER053
 Description: Driver cl29.1
 Directory: c:\29.1.log
 Machine: cl29.hp-perf.net
 Parameter Set: Optimized
 Index: 520000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER53605203890
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25

CLIENT_NURAND: 233
 Additional Options:

 Name: DRIVER054
 Description: Driver cl29.2
 Directory: c:\29.2.log
 Machine: cl29.hp-perf.net
 Parameter Set: Optimized
 Index: 530000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER54605223093
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

 Name: DRIVER055
 Description: Driver cl30.1
 Directory: c:\30.1.log
 Machine: cl30.hp-perf.net
 Parameter Set: Optimized
 Index: 540000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER55605251687
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

 Name: DRIVER056
 Description: Driver cl30.2
 Directory: c:\30.2.log
 Machine: cl30.hp-perf.net
 Parameter Set: Optimized
 Index: 550000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER56605282625
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

Name: DRIVER057
 Description: Driver cl31.1
 Directory: c:\31.1.log
 Machine: cl31.hp-perf.net
 Parameter Set: Optimized
 Index: 560000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER57606847843
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

 Name: DRIVER058
 Description: Driver cl31.2
 Directory: c:\31.2.log
 Machine: cl31.hp-perf.net
 Parameter Set: Optimized
 Index: 570000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER58607086906
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

 Name: DRIVER059
 Description: Driver cl32.1
 Directory: c:\32.1.log
 Machine: cl32.hp-perf.net
 Parameter Set: Optimized
 Index: 580000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER59607716484
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

 Name: DRIVER060
 Description: Driver cl32.2

Directory: c:\32.2.log
Machine: cl32.hp-perf.net
Parameter Set: Optimized
Index: 590000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER60607735531
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER061
Description: Driver cl33.1
Directory: c:\33.1.log
Machine: cl33.hp-perf.net
Parameter Set: Optimized
Index: 600000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER61607933578
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER062
Description: Driver cl33.2
Directory: c:\33.2.log
Machine: cl33.hp-perf.net
Parameter Set: Optimized
Index: 610000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER62607953015
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER063
Description: Driver cl34.1
Directory: c:\34.1.log

Parameter Set: Optimized
Index: 620000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER63607975265
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER064
Description: Driver cl34.2
Directory: c:\34.2.log
Machine: cl34.hp-perf.net
Parameter Set: Optimized
Index: 630000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER64607993515
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER065
Description: Driver cl35.1
Directory: c:\35.1.log
Machine: cl35.hp-perf.net
Parameter Set: Optimized
Index: 640000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER65608223078
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER066
Description: Driver cl35.2
Directory: c:\35.2.log
Machine: cl35.hp-perf.net
Parameter Set: Optimized
Index: 650000000

Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER66608251125
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER067
Description: Driver cl36.1
Directory: c:\36.1.log
Machine: cl36.hp-perf.net
Parameter Set: Optimized
Index: 660000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER67608277640
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER068
Description: Driver cl36.2
Directory: c:\36.2.log
Machine: cl36.hp-perf.net
Parameter Set: Optimized
Index: 670000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER68608297015
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER069
Description: Driver cl37.1
Directory: c:\37.1.log
Machine: cl37.hp-perf.net
Parameter Set: Optimized
Index: 680000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER69611194234

Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

 Name: DRIVER070
 Description: Driver cl37.2
 Directory: c:\37.2.log
 Machine: cl37.hp-perf.net
 Parameter Set: Optimized
 Index: 690000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER70611233406
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

 Name: DRIVER071
 Description: Driver cl38.1
 Directory: c:\38.1.log
 Machine: cl38.hp-perf.net
 Parameter Set: Optimized
 Index: 700000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER71611256781
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

 Name: DRIVER072
 Description: Driver cl38.2
 Directory: c:\38.2.log
 Machine: cl38.hp-perf.net
 Parameter Set: Optimized
 Index: 710000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER72611279406
 Connect Rate: 500

Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

 Name: DRIVER073
 Description: Driver cl39.1
 Directory: c:\39.1.log
 Machine: cl39.hp-perf.net
 Parameter Set: Optimized
 Index: 720000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER73611801359
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

 Name: DRIVER074
 Description: Driver 39.2
 Directory: c:\39.2.log
 Machine: cl39.hp-perf.net
 Parameter Set: Optimized
 Index: 730000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER74611825281
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

 Name: DRIVER075
 Description: Driver cl40.1
 Directory: c:\40.1.log
 Machine: cl40.hp-perf.net
 Parameter Set: Optimized
 Index: 740000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER75611844781
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25

CLIENT_NURAND: 233
 Additional Options:

 Name: DRIVER076
 Description: Driver cl40.2
 Directory: c:\40.2.log
 Machine: cl40.hp-perf.net
 Parameter Set: Optimized
 Index: 750000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER76611871312
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

 Name: DRIVER077
 Description: Driver cl41.1
 Directory: c:\41.1.log
 Machine: cl41.hp-perf.net
 Parameter Set: Optimized
 Index: 760000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER77623019031
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

 Name: DRIVER078
 Description: Driver cl41.2
 Directory: c:\41.2.log
 Machine: cl41.hp-perf.net
 Parameter Set: Optimized
 Index: 770000000
 Seed: 69483
 Configured Users: 5130
 Pipe Name: DRIVER78623042718
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

Name: DRIVER079
Description: Driver cl42.1
Directory: c:\42.1.log
Machine: cl42.hp-perf.net
Parameter Set: Optimized
Index: 780000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER79623072734
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER080
Description: Driver cl42.2
Directory: c:\42.2.log
Machine: cl42.hp-perf.net
Parameter Set: Optimized
Index: 790000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER80623093796
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER081
Description: Driver cl43.1
Directory: c:\43.1.log
Machine: cl43.hp-perf.net
Parameter Set: Optimized
Index: 800000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER81623264640
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER082
Description: Driver cl43.2
Directory: c:\43.2.log
Machine: cl43.hp-perf.net
Parameter Set: Optimized
Index: 810000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER82623291484
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER083
Description: Driver cl44.1
Directory: c:\44.1.log
Machine: cl44.hp-perf.net
Parameter Set: Optimized
Index: 820000000
Seed: 69483
Configured Users: 5260
Pipe Name: DRIVER83623312984
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER084
Description: Driver cl44.2
Directory: c:\44.2.log
Machine: cl44.hp-perf.net
Parameter Set: Optimized
Index: 830000000
Seed: 69483
Configured Users: 5260
Pipe Name: DRIVER84623335625
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER085
Description: Driver cl45.1
Directory: c:\45.1.log

Name: DRIVER086
Description: Driver cl45.2
Directory: c:\45.2.log
Machine: cl45.hp-perf.net
Parameter Set: Optimized
Index: 840000000
Seed: 69483
Configured Users: 5260
Pipe Name: DRIVER85623502656
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER087
Description: Driver cl46.1
Directory: c:\46.1.log
Machine: cl46.hp-perf.net
Parameter Set: Optimized
Index: 850000000
Seed: 69483
Configured Users: 5260
Pipe Name: DRIVER86623525546
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER087
Description: Driver cl46.1
Directory: c:\46.1.log
Machine: cl46.hp-perf.net
Parameter Set: Optimized
Index: 860000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER87623548312
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER088
Description: Driver cl46.2
Directory: c:\46.2.log
Machine: cl46.hp-perf.net
Parameter Set: Optimized
Index: 870000000

Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER88623574468
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER089
Description: Driver cl47.1
Directory: c:\47.1.log
Machine: cl47.hp-perf.net
Parameter Set: Optimized
Index: 880000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER89624090859
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER090
Description: Driver cl47.2
Directory: c:\47.2.log
Machine: cl47.hp-perf.net
Parameter Set: Optimized
Index: 890000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER90624115062
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER091
Description: Driver cl48.1
Directory: c:\48.1.log
Machine: cl48.hp-perf.net
Parameter Set: Optimized
Index: 900000000
Seed: 69483

Pipe Name: DRIVER91624146890
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER092
Description: Driver cl48.2
Directory: c:\48.2.log
Machine: cl48.hp-perf.net
Parameter Set: Optimized
Index: 910000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER92624377421
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER093
Description: Driver cl49.1
Directory: c:\49.1.log
Machine: cl49.hp-perf.net
Parameter Set: Optimized
Index: 920000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER93624628406
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER094
Description: Driver cl49.2
Directory: c:\49.2.log
Machine: cl49.hp-perf.net
Parameter Set: Optimized
Index: 930000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER94624649421
Connect Rate: 500

Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER095
Description: Driver cl50.1
Directory: c:\50.1.log
Machine: cl50.hp-perf.net
Parameter Set: Optimized
Index: 940000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER95624668875
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER096
Description: Driver cl50.2
Directory: c:\50.2.log
Machine: cl50.hp-perf.net
Parameter Set: Optimized
Index: 950000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER96624698359
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER097
Description: Driver cl51.1
Directory: c:\51.1.log
Machine: cl51.hp-perf.net
Parameter Set: Optimized
Index: 960000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER97624720968
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25

CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER098
Description: Driver cl51.2
Directory: c:\51.2.log
Machine: cl51.hp-perf.net
Parameter Set: Optimized
Index: 970000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER98624741531
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER099
Description: Driver cl52.1
Directory: c:\52.1.log
Machine: cl52.hp-perf.net
Parameter Set: Optimized
Index: 980000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER99624773593
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER100
Description: Driver cl52.2
Directory: c:\52.2.log
Machine: cl52.hp-perf.net
Parameter Set: Optimized
Index: 990000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER100624795109
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233

Additional Options:

Name: DRIVER101
Description: Driver cl53.1
Directory: c:\53.1.log
Machine: cl53.hp-perf.net
Parameter Set: Optimized
Index: 1000000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER101624822937
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER102
Description: Driver cl53.2
Directory: c:\53.2.log
Machine: cl53.hp-perf.net
Parameter Set: Optimized
Index: 1010000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER102625305796
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER103
Description: Driver cl54.1
Directory: c:\54.1.log
Machine: cl54.hp-perf.net
Parameter Set: Optimized
Index: 1020000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER103625374875
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER104
Description: Driver cl54.2
Directory: c:\54.2.log
Machine: cl54.hp-perf.net
Parameter Set: Optimized
Index: 1030000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER104625592843
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER105
Description: Driver cl55.1
Directory: c:\55.1.log
Machine: cl55.hp-perf.net
Parameter Set: Optimized
Index: 1040000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER105625932718
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER106
Description: Driver cl55.2
Directory: c:\55.2.log
Machine: cl55.hp-perf.net
Parameter Set: Optimized
Index: 1050000000
Seed: 69483
Configured Users: 5130
Pipe Name: DRIVER106625953750
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER107
Description: Driver cl56.1
Directory: c:\56.1.log

Machine: cl56.hp-perf.net
Parameter Set: Optimized
Index: 1060000000
Seed: 69483
Configured Users: 5020
Pipe Name: DRIVER107626015156
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER108
Description: Driver cl56.2
Directory: c:\56.2.log
Machine: cl56.hp-perf.net
Parameter Set: Optimized
Index: 1070000000
Seed: 69483
Configured Users: 5020
Pipe Name: DRIVER108626053421
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER109
Description: Driver cl57.1
Directory: c:\57.1.log
Machine: cl57.hp-perf.net
Parameter Set: Optimized
Index: 1080000000
Seed: 69483
Configured Users: 5020
Pipe Name: DRIVER109626376921
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER110
Description: Driver cl57.2
Directory: c:\57.2.log
Machine: cl57.hp-perf.net

Name: DRIVER109
Description: Driver cl58.1
Directory: c:\58.1.log
Machine: cl58.hp-perf.net
Parameter Set: Optimized
Index: 1090000000
Seed: 69483
Configured Users: 5020
Pipe Name: DRIVER110626396906
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER111
Description: Driver cl58.1
Directory: c:\58.1.log
Machine: cl58.hp-perf.net
Parameter Set: Optimized
Index: 1100000000
Seed: 69483
Configured Users: 5000
Pipe Name: DRIVER111626420093
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER112
Description: Driver cl58.2
Directory: c:\58.2.log
Machine: cl58.hp-perf.net
Parameter Set: Optimized
Index: 1110000000
Seed: 69483
Configured Users: 5000
Pipe Name: DRIVER112626442078
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER113
Description: Driver cl59.1
Directory: c:\59.1.log
Machine: cl59.hp-perf.net
Parameter Set: Optimized
Index: 1120000000
Seed: 69483

Pipe Name: DRIVER113627277406
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER114
Description: Driver cl59.2
Directory: c:\59.2.log
Machine: cl59.hp-perf.net
Parameter Set: Optimized
Index: 1130000000
Seed: 69483
Configured Users: 5000
Pipe Name: DRIVER114627299609
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER115
Description: Driver cl60.1
Directory: c:\60.1.log
Machine: cl60.hp-perf.net
Parameter Set: Optimized
Index: 1140000000
Seed: 69483
Configured Users: 5160
Pipe Name: DRIVER115627322015
Connect Rate: 500
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 25
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER116
Description: Driver cl60.2
Directory: c:\60.2.log
Machine: cl60.hp-perf.net
Parameter Set: Optimized
Index: 1150000000
Seed: 69483
Configured Users: 5160
Pipe Name: DRIVER116627344250
Connect Rate: 500

Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

 Name: DRIVER117
 Description: Driver cl61.1
 Directory: c:\61.1.log
 Machine: cl61.hp-perf.net
 Parameter Set: Optimized
 Index: 1160000000
 Seed: 69483
 Configured Users: 5160
 Pipe Name: DRIVER117627539515
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

 Name: DRIVER118
 Description: Driver cl61.2
 Directory: c:\61.2.log
 Machine: cl61.hp-perf.net
 Parameter Set: Optimized
 Index: 1170000000
 Seed: 69483
 Configured Users: 5160
 Pipe Name: DRIVER118627570937
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

 Name: DRIVER119
 Description: Driver cl62.1
 Directory: c:\62.1.log
 Machine: cl62.hp-perf.net
 Parameter Set: Optimized
 Index: 1180000000
 Seed: 69483
 Configured Users: 5200
 Pipe Name: DRIVER119627943687
 Connect Rate: 500
 Start Rate: 150

~~Max. Concurrency Rate: 25~~
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

 Name: DRIVER120
 Description: Driver cl62.2
 Directory: c:\62.2.log
 Machine: cl62.hp-perf.net
 Parameter Set: Optimized
 Index: 1190000000
 Seed: 69483
 Configured Users: 5200
 Pipe Name: DRIVER120627959500
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

 Name: DRIVER121
 Description: Driver cl63.1
 Directory: c:\63.1.log
 Machine: cl63.hp-perf.net
 Parameter Set: Optimized
 Index: 1200000000
 Seed: 69483
 Configured Users: 5200
 Pipe Name: DRIVER121627977703
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

 Name: DRIVER122
 Description: Driver cl63.2
 Directory: c:\63.2.log
 Machine: cl63.hp-perf.net
 Parameter Set: Optimized
 Index: 1210000000
 Seed: 69483
 Configured Users: 5200
 Pipe Name: DRIVER122627997906
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233

Additional Options:

 Name: DRIVER123
 Description: Driver cl2.1
 Directory: c:\2.1.log
 Machine: cl2.hp-perf.net
 Parameter Set: Optimized
 Index: 1220000000
 Seed: 69483
 Configured Users: 5200
 Pipe Name: DRIVER1271445497937
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

 Name: DRIVER124
 Description: Driver cl2.2
 Directory: c:\2.2.log
 Machine: cl2.hp-perf.net
 Parameter Set: Optimized
 Index: 1230000000
 Seed: 69483
 Configured Users: 5200
 Pipe Name: DRIVER1281445547656
 Connect Rate: 500
 Start Rate: 150
 Max. Concurrency: -1
 Concurrency Rate: 25
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

 Number of User groups: 124

 Driver Engine: DRIVER001
 IIS Server: w1.hp-perf.net
 SQL Server: via:sqlorca
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 1 - 520
 w_id Min Warehouse: 1
 w_id Max Warehouse: 63592
 Scale: Normal
 User Count: 5200
 District id: 1
 Scale Down: No

 Driver Engine: DRIVER002

IIS Server: w1.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 521 - 1040
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5200
District id: 1
Scale Down: No

Driver Engine: DRIVER003
IIS Server: w1.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1041 - 1560
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5200
District id: 1
Scale Down: No

Driver Engine: DRIVER004
IIS Server: w2.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1561 - 2080
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5200
District id: 1
Scale Down: No

Driver Engine: DRIVER005
IIS Server: w2.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2081 - 2600
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5200

Scale Down: No

Driver Engine: DRIVER006
IIS Server: w2.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2601 - 3120
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5200
District id: 1
Scale Down: No

Driver Engine: DRIVER007
IIS Server: w3.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3121 - 3615
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 4950
District id: 1
Scale Down: No

Driver Engine: DRIVER008
IIS Server: w3.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3616 - 4110
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 4950
District id: 1
Scale Down: No

Driver Engine: DRIVER009
IIS Server: w3.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4111 - 4605
w_id Min Warehouse: 1
w_id Max Warehouse: 63592

Scale Down: No
User Count: 4950
District id: 1
Scale Down: No

Driver Engine: DRIVER010
IIS Server: w3.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4606 - 5100
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 4950
District id: 1
Scale Down: No

Driver Engine: DRIVER011
IIS Server: w4.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5101 - 5613
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER012
IIS Server: w4.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5614 - 6126
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER013
IIS Server: w4.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 6127 - 6639

w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER014
IIS Server: w4.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 6640 - 7152
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER015
IIS Server: w5.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 7153 - 7665
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER016
IIS Server: w5.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 7666 - 8178
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER017
IIS Server: w5.hp-perf.net
SQL Server: via:sqlorca

Database: tpcc
Protocol: HTML
w_id Range: 8179 - 8691
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER018
IIS Server: w5.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 8692 - 9204
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER019
IIS Server: w6.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 9205 - 9717
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER020
IIS Server: w6.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 9718 - 10230
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER021

SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 10231 - 10743
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER022
IIS Server: w6.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 10744 - 11256
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER023
IIS Server: w7.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 11257 - 11769
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER024
IIS Server: w7.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 11770 - 12282
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER025
IIS Server: w7.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 12283 - 12795
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER026
IIS Server: w7.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 12796 - 13308
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER027
IIS Server: w8.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 13309 - 13821
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER028
IIS Server: w8.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 13822 - 14334
w_id Min Warehouse: 1
w_id Max Warehouse: 63592

Driver Engine: DRIVER029
IIS Server: w8.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 14335 - 14847
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER030
IIS Server: w8.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 14848 - 15360
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER031
IIS Server: w9.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 15361 - 15873
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER032
IIS Server: w9.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 15874 - 16386

Driver Engine: DRIVER033
IIS Server: w9.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 16387 - 16899
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER034
IIS Server: w9.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 16900 - 17412
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER035
IIS Server: w10.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 17413 - 17925
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER036
IIS Server: w10.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa

Protocol: HTML
w_id Range: 17926 - 18438
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER037
IIS Server: w10.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 18439 - 18951
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER038
IIS Server: w10.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 18952 - 19464
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER039
IIS Server: w11.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 19465 - 19977
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER040

SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 19978 - 20490
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER041
IIS Server: w11.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 20491 - 21003
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER042
IIS Server: w11.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 21004 - 21516
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER043
IIS Server: w12.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 21517 - 22029
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER044
IIS Server: w12.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 22030 - 22542
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER045
IIS Server: w12.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 22543 - 23055
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER046
IIS Server: w12.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 23056 - 23568
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER047
IIS Server: w13.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 23569 - 24081
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130

District id: 1
Scale Down: No

Driver Engine: DRIVER048
IIS Server: w13.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 24082 - 24594
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER049
IIS Server: w13.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 24595 - 25107
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER050
IIS Server: w13.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 25108 - 25620
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER051
IIS Server: w14.hp-perf.net
SQL Server: via:sqlserver
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 25621 - 26133

w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER052
IIS Server: w14.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 26134 - 26646
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER053
IIS Server: w14.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 26647 - 27159
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER054
IIS Server: w14.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 27160 - 27672
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER055
IIS Server: w15.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa

w_id Range: 27673 - 28185
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER056
IIS Server: w15.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 28186 - 28698
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER057
IIS Server: w15.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 28699 - 29211
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER058
IIS Server: w15.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 29212 - 29724
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER059
IIS Server: w16.hp-perf.net
SQL Server: via:sqlorca

Database: tpcc
User: sa
Protocol: HTML
w_id Range: 29725 - 30237
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER060
IIS Server: w16.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 30238 - 30750
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER061
IIS Server: w16.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 30751 - 31263
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER062
IIS Server: w16.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 31264 - 31776
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER063
IIS Server: w17.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 31777 - 32289
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER064
IIS Server: w17.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 32290 - 32802
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER065
IIS Server: w17.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 32803 - 33315
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER066
IIS Server: w17.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 33316 - 33828
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130

Scale Down: No

Driver Engine: DRIVER067
IIS Server: w18.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 33829 - 34341
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER068
IIS Server: w18.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 34342 - 34854
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER069
IIS Server: w18.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 34855 - 35367
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER070
IIS Server: w18.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 35368 - 35880
w_id Min Warehouse: 1
w_id Max Warehouse: 63592

Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER071
IIS Server: w19.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 35881 - 36393
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER072
IIS Server: w19.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 36394 - 36906
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER073
IIS Server: w19.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 36907 - 37419
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER074
IIS Server: w19.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa

Driver Engine: DRIVER075
IIS Server: w20.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 37933 - 38445
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER076
IIS Server: w20.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 38446 - 38958
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER077
IIS Server: w20.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 38959 - 39471
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER078
IIS Server: w20.hp-perf.net
SQL Server: via:sqlorca

Driver Engine: DRIVER079
IIS Server: w21.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 39472 - 39984
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER080
IIS Server: w21.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 40497 - 40985
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER081
IIS Server: w21.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 40986 - 41500
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER082
IIS Server: w21.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 41501 - 42015
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

IIS Server: w21.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 41524 - 42036
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER083
IIS Server: w22.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 42037 - 42562
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5260
District id: 1
Scale Down: No

Driver Engine: DRIVER084
IIS Server: w22.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 42563 - 43088
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5260
District id: 1
Scale Down: No

Driver Engine: DRIVER085
IIS Server: w22.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 43089 - 43614
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5260

Scale Down: No

Driver Engine: DRIVER086
IIS Server: w22.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 43615 - 44140
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5260
District id: 1
Scale Down: No

Driver Engine: DRIVER087
IIS Server: w23.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 44141 - 44653
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER088
IIS Server: w23.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 44654 - 45166
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER089
IIS Server: w23.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 45167 - 45679
w_id Min Warehouse: 1
w_id Max Warehouse: 63592

Scale Down: No
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER090
IIS Server: w23.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 45680 - 46192
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER091
IIS Server: w24.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 46193 - 46705
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER092
IIS Server: w24.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 46706 - 47218
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER093
IIS Server: w24.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 47219 - 47731

w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER094
IIS Server: w24.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 47732 - 48244
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER095
IIS Server: w25.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 48245 - 48757
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER096
IIS Server: w25.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 48758 - 49270
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER097
IIS Server: w25.hp-perf.net
SQL Server: via:sqlorca

Database: tpcc
Protocol: HTML
w_id Range: 49271 - 49783
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER098
IIS Server: w25.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 49784 - 50296
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER099
IIS Server: w26.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 50297 - 50809
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER100
IIS Server: w26.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 50810 - 51322
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER101

SQL Server: w26.hp-perf.net
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 51323 - 51835
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER102
IIS Server: w26.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 51836 - 52348
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER103
IIS Server: w27.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 52349 - 52861
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER104
IIS Server: w27.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 52862 - 53374
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER105
IIS Server: w27.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 53375 - 53887
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER106
IIS Server: w27.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 53888 - 54400
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5130
District id: 1
Scale Down: No

Driver Engine: DRIVER107
IIS Server: w28.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 54401 - 54902
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5020
District id: 1
Scale Down: No

Driver Engine: DRIVER108
IIS Server: w28.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 54903 - 55404
w_id Min Warehouse: 1
w_id Max Warehouse: 63592

User Count: 5020
District id: 1
Scale Down: No

Driver Engine: DRIVER109
IIS Server: w28.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 55405 - 55906
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5020
District id: 1
Scale Down: No

Driver Engine: DRIVER110
IIS Server: w28.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 55907 - 56408
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5020
District id: 1
Scale Down: No

Driver Engine: DRIVER111
IIS Server: w29.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 56409 - 56908
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5000
District id: 1
Scale Down: No

Driver Engine: DRIVER112
IIS Server: w29.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 56909 - 57408

w_id Max Warehouse: 63592
Scale: Normal
User Count: 5000
District id: 1
Scale Down: No

Driver Engine: DRIVER113
IIS Server: w29.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 57409 - 57908
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5000
District id: 1
Scale Down: No

Driver Engine: DRIVER114
IIS Server: w29.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 57909 - 58408
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5000
District id: 1
Scale Down: No

Driver Engine: DRIVER115
IIS Server: w30.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 58409 - 58924
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER116
IIS Server: w30.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa

Protocol: HTML
w_id Range: 58925 - 59440
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER117
IIS Server: w30.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 59441 - 59956
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER118
IIS Server: w30.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 59957 - 60472
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER119
IIS Server: w31.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 60473 - 60992
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5200
District id: 1
Scale Down: No

Driver Engine: DRIVER120

SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 60993 - 61512
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5200
District id: 1
Scale Down: No

Driver Engine: DRIVER121
IIS Server: w31.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 61513 - 62032
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5200
District id: 1
Scale Down: No

Driver Engine: DRIVER122
IIS Server: w32.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 62033 - 62552
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5200
District id: 1
Scale Down: No

Driver Engine: DRIVER123
IIS Server: w32.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 62553 - 63072
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5200
District id: 1
Scale Down: No

Driver Engine: DRIVER124
IIS Server: w32.hp-perf.net
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 63073 - 63592
w_id Min Warehouse: 1
w_id Max Warehouse: 63592
Scale: Normal
User Count: 5200
District id: 1
Scale Down: No

Number of Parameter Sets: 3

		~Default				
		Default Parameter Set				
		Txn	Think	Key	RT	RT
Menu		Weight	Time	Time	Delay	Fence
Delay		New Order	10.00	12.05	18.01	
	0.10	5.00	0.10			
		Payment	10.00	12.05	3.01	
	0.10	5.00	0.10			
		Delivery	1.00	5.05	2.01	
	0.10	5.00	0.10			
		Stock Level	1.00	5.05	2.01	
	0.10	20.00	0.10			
		Order Status	1.00	10.05	2.01	
	0.10	5.00	0.10			
		Optimized				
		Txn	Think	Key	RT	RT
Menu		Weight	Time	Time	Delay	Fence
Delay		New Order	44.95	12.05	18.01	
	0.10	5.00	0.10			
		Payment	43.01	12.05	3.01	
	0.10	5.00	0.10			
		Delivery	4.01	5.05	2.01	
	0.10	5.00	0.10			
		Stock Level	4.02	5.05	2.01	
	0.10	20.00	0.10			
		Order Status	4.01	10.05	2.01	
	0.10	5.00	0.10			
		shutdown				

Menu	Txn Think Key RT RT				
	Weight	Time	Time	Delay	Fence
Delay		New Order	0.00	12.05	18.01
	0.10	5.00	0.10		
		Payment	0.00	12.05	3.01
	0.10	5.00	0.10		
		Delivery	0.00	5.05	2.01
	0.10	5.00	0.10		
		Stock Level	0.00	5.05	2.01
	0.10	20.00	0.10		

Appendix D 60 Day Space Requirements

TPC-C 60 Day Space Requirements						
Warehouses	72,000				TpmC	786,646.62
Table	Rows	Data KB	Index KB	Extra 5% KB	8hr Space	Total Space KB
Warehouse	72,000	7,888	600	424		8912
District	720,000	81,216	720	4,097		86033
Customer	2,147,483,647	1,570,909,176	98,141,752	83,452,546		1752503474
History	2,147,483,647	128,955,312	512,816		110,127,254	129468128
New_order	648,000,000	11,545,752	38,736	579,224		12163712
Orders	2,147,483,647	70,530,704	171,088		238,103,050	70701792
Order_line	2,147,483,647	1,440,005,288	3,492,536		1,098,749,337	1443497824
Item	100,000	9,528	544	504		10576
Stock	2,147,483,647	2,304,000,008	4,979,776	115,448,989		2424428773
Total		5,526,044,872	107,338,568	199,485,785	1,446,979,641	5,832,869,225
MB						
Dynamic Space	1,601,066	Sum of Data for Order, Orderline and History				
Static Space	4,095,096	Sum of Data+Index+5%-Dynamic Space				
Free Space	na	Total Allocated Spac - (Dynamic + Static Space)				
Daily Growth	279,883	(Dynamic Space/(W*62.5))*tpmc				
Daily Spread	-	(Free Space -1.5*Daily Growth) Zero Assumed				
60 Day Space MB	20,888,068					
60 Day Space GB	20,398.50	GB				
Log Size	800,000.00	MB				
KB Per New Order	5.47	KB				
8 hr log MB	2,017,011	MB				
8 hr log GB	1,969.7374	GB				
Space Usage	GB Needed	Disks Measured	GB Priced	Disk Size	Formatted Size	
60 Day Space DB	20,399	1792	30,285	18.2GB	16.900	
			0.00			
			0.00			
Total DB			30284.80			
8-hr log + mirror	3,939	60	4,269	73GB	71.15	
OS, Swap	3	2	72.00	18.2GB	36.000	
Total Storage	24,340.98	GB	34,625.80	GB		

Appendix E 3rd Party Pricing

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
<http://www.microsoft.com/>

Microsoft

August 25, 2003

Hewlett-Packard
Company
Gunter Zink
14335 NE 24th St
Bellevue, WA 98005

Mr. Zink:

Here is the information you requested regarding pricing for several Microsoft products to be used in conjunction with your TPC-C benchmark testing.

All pricing shown is in US Dollars (\$).

Part Number	Description	Unit Price	Quantity	Price
810-00560	SQL Server 2000 Enterprise Edition (64-bit) <i>Per processor licensing Discount Schedule: Open Program Level C Unit Price reflects a 17% discount from the retail unit price of \$19,999.</i>	\$16,541	64	\$1,058,624
C11-00821	Windows 2000 Server <i>Server license only - No CALs Discount Schedule: Open Program - No Level Unit Price reflects a 8% discount from the retail unit price of \$799.</i>	\$738	32	\$23,616
254-00170	Visual C++ Standard <i>No discounts applied</i>	\$109	1	\$109
PRO-PRORS-16U-01	Database Server Support Package <i>1 Year Term</i>	\$1,950	3	\$5,850

All products are currently orderable through Microsoft's normal distribution channels.

This quote is valid for the next 90 days.

If we can be of any further assistance, please contact Jamie Reding at (425) 703-0510 or jamiere@microsoft.com.

Reference ID: PCguzi0325087290

Please include this Reference ID in any correspondence regarding this price quote.

Aug 20, 2003

Larry Kemp
Hewlett-Packard Company

Subject: Qlogic MSRP for TPC Benchmark Publication

Dear Larry,

Qlogic is pleased to provide you the following MSRP quote for your TPC benchmark publication.
The following prices are effective through Dec 31, 2003.

Product	Distributor(s)	List Price
QLA2350 1-port FC HBA	Direct from QLogic	\$1,765
SANBox 2-16 FC Switch	Unique, Bell, Tech Data, Arrow	\$15,995

Best regards,

Russ Logan
Senior OEM Sales Manager
(512) 289.2619
Russell.logan@qlogic.com