



TPC Benchmark® C

Full Disclosure Report

HP Server rx8620

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

Fourth Edition
April 13, 2004

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 2004.

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., April 13, 2004

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

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

Intel, Pentium, Xeon and Itanium 2 are registered trademarks of the Intel 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 Server rx8620 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

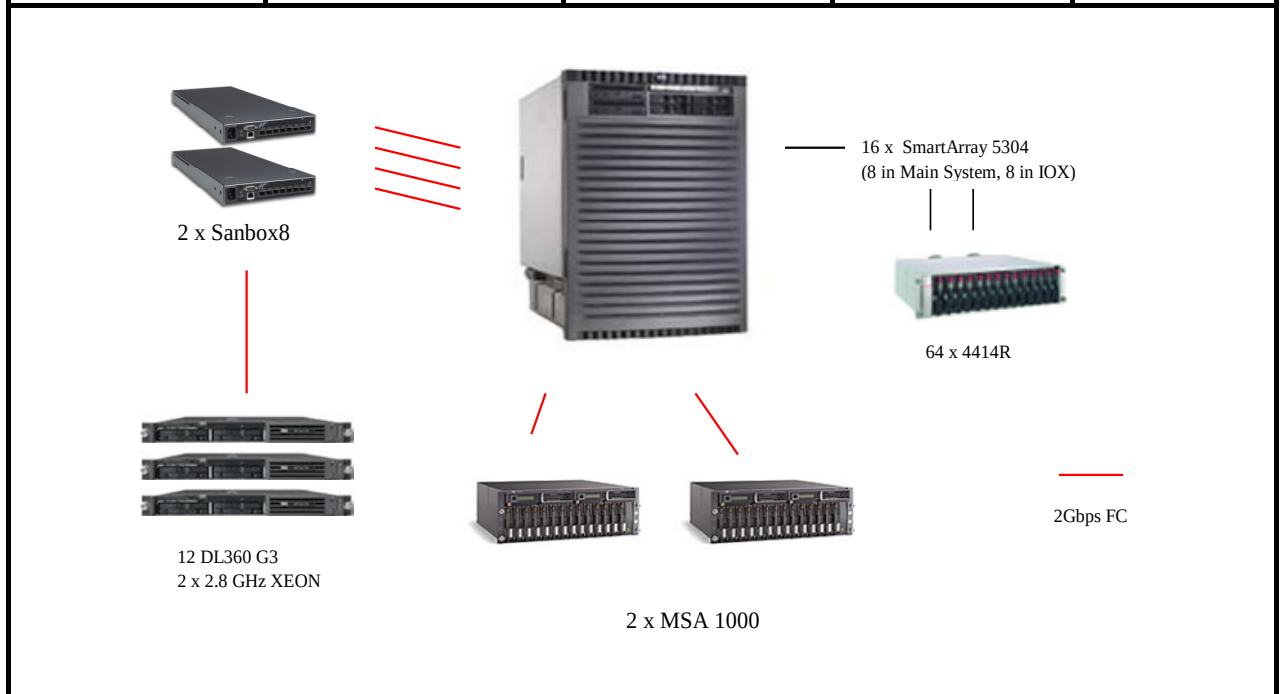
The following pages contain the executive summary of the benchmark results for the HP Server rx8620 system. The Standard System Summary is given below.

Company Name	System Name	Database Software	Operating System
Hewlett-Packard Company	HP Server rx8620	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
\$1,372,435	301225.52 tpmC	\$4.56 per tpmC	April 15, 2004

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 Server rx8620			TPC-C Revision 5.1
Total System Cost	TPC Throughput	Price/Performance	Report Date	Availability Date
\$1,372,435	301225.52 tpmC	\$4.56 per tpmC	April 13, 2004	April 15, 2004
Processors	Database Manager	Operating System	Other Software	Number of Users
16 Intel Itanium2 1.5GHz - Server 24 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	241,920



System Components	Server		Each Client	
	Qty	Type	Qty	Type
Processors	16	Intel Itanium 2 6M CPUs at 1.5GHz	2	2.8 GHz Intel Xeon
Cache Memory		6 MB iL3 cache		512kbyte L2 Cache
Memory	16	8 Gbyte	4	256 Mbyte
Disk Controllers	16	HP Smart Array 5304	1	Embedded Smart Array 5i+
Disk Drives	896	HP 18GB 15KRPM U320	1	36 Gbyte disk
	12	HP 146GB, 10krpm U320		
	2	HP 18GB 15KRPM for OS		
Total Storage		16807 Gbyte		36 Gbyte
Tape Drives	1	HP Surestore DAT40e		
Terminals	1	Console Terminal	1	Console Terminal

 Hewlett Packard Company		HP Server rx8620		TPC-C Rev. 5.1		
				Report Date: April 13, 2004		
Description	Price Key	Part Number	Unit Price	Qty	Extended Price	3 Yr Maint Price
rx8620 16-way SMP Base System incl cell boards, core IO, power supplies, PCI-X backplan, 16x1.5ghz Intel Itanium 2 6M processors	1	AB240A-002	\$273,995	1	\$273,995	
Support (Hardware & Software) 3yrs 24x7					\$0	\$117,292
8GB SDRAM (4x2GB DIMMs)	1	AB309A-OD1	\$14,000	16	\$224,000	
PCI-x I/O chassis	1	A6434A	\$29,995	1	\$29,995	
18gb 15K RPM disc for system boot/storage	1	A6540A	\$800	2	\$1,600	
Gig E - TX Adapter	1	A7061A	\$270	1	\$270	
QLogic QLA-2350 Fibre-Channel VI Adapter	1	A390975	\$1,450	6	\$8,700	
LP9802 Fiber Channel Adapter	1	AB232A	\$1,558	2	\$3,116	
HP Smart Array Controller 5304	1	A9826A	\$2,399	16	\$38,384	
Storageworks enclosure 4414R	1	302969-B21	\$3,209	64	\$205,376	
Support 3yrs 24x7	1	171242-002	\$157	64		\$10,048
18GB, 15krpm Ultra3 Wide disk	1	286775-B22	\$299	896	\$267,904	
18GB, 15krpm Ultra3 Wide disk (10% spares)	1	286775-B22	\$299	90	\$26,910	
Modular Storage Array 1000	1	201723-B22	\$9,995	2	\$19,990	
Support 3yrs 24x7	1	402164-002	\$3,538	2		\$7,076
146GB, 10krpm disk	1	286716-B22	\$1,039	12	\$12,468	
146GB, 10krpm disk (10% spares)	1	286716-B22	\$1,039	2	\$2,078	
HP9000 Standard Rack System E41	1	A4902D	\$1,910	5	\$9,550	
5m LC to LC Cable Kit	1	221692-B22	\$82	2	\$164	
5m 2gbt Small Form Pluggable Adapter Kit	1	221470-B21	\$199	2	\$398	
HP Power Distribution Unit 120-240V	1	A5137AZ	\$145	6	\$870	
UPS R1500 XR	1	204404-001	\$866	1	\$866	
TA5300 Enclosure for DAT tape	1	C7508AZ	\$1,395	1	\$1,395	\$2,125
DVD Rom drive	1	C7499A	\$688	1	\$688	
DAT Tape	1	C7497A	\$1,532	1	\$1,532	
Server Subtotal					\$1,130,249	\$136,541
Microsoft Windows Server 2003, Datacenter edition (64-bit)	1	T2372A opt016	\$32,000	1	\$32,000 incl. above	
Database Server Support Package	1	PRO-PRORS-16U-01	\$1,950	3	\$5,850	\$5,850
Microsoft SQL Server 2000 Enterprise Edition 64 bit	2	810-00560	\$16,541	16	\$264,656	
Server Software Subtotal					\$302,506	\$5,850
HP ProLiant DL360 G3	1	308724-001	\$2,849.00	12	\$34,188	
2.80GHz Xeon Processor Option Kit (DL360 G3)	1	292892-B21	\$839.00	12	\$10,068	
HP Mouse	1	170299-B22	\$23.00	1	\$23	
HP Enhanced Keyboard	1	265977-001	\$12.00	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	\$519	12	\$6,228	
Qlogic QLA-2350 Fibre-Channel VI Adapter	1	A390975	\$1,450	12	\$17,400	
5M LC to LC Cable Kit	1	221692-B22	\$82	12	\$984	
5m 2gbt Small Form Pluggable Adapter Kit	1	221470-B21	\$199	12	\$2,388	
Support (Hardware & Software) 3yrs 24x7						\$34,104
Client Subtotal					\$71,420	\$34,104
Microsoft Windows 2000 Server	2	C11-00821	\$738	12	\$8,856 incl. above	
Microsoft Visual C++ Standard 6.0	2	254-0017D	\$109	1	\$109	\$1,009
Client Software Subtotal					8965	\$1,009
Qlogic SANBox-2 8-Port Switch	3	SANBOX 2/8	\$4,861	4	\$19,444	
Connectivity Subtotal					\$19,444	
HP's Large Configuration Discount *	* Discounts:				-\$297,429	-\$40,224
Total:					\$1,235,155	\$137,280
Price Key: 1-HP, 2-Microsoft, 3-PC Universe Audited by Lorna Livingtree of Performance Metrics, Inc.		3 year cost of ownership:				\$1,372,435
		tpmC:				301,225.52
* All discounts are based on US list prices and for similar quantities and configurations		\$/tpmC:				\$4.56
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 Server rx8620

MQTH, Computed Maximum Qualified Throughput

301225.52 tpmC

Response Times (in seconds)

	90th %-ile	Maximum	Average
New-Order	0.90s	17.78s	0.54s
Payment	0.84s	56.09s	0.48s
Order-Status	0.87s	13.57s	0.52s
Delivery (interactive portion)	0.11s	1.20s	0.10s
Delivery (deferred portion)	0.19s	3.88s	0.14s
Stock-Level	1.59s	16.74s	1.08s
Menu	0.11s	6.46s	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	100.5s
Delivery (interactive)	2s	2.02s	2.03s	0s	5.05s	50.51s
Stock-Level	2s	2.02s	2.03s	0s	5.06s	50.5s

Test Duration

Ramp up time	45 minutes
Measurement interval	120 minutes
Transactions during measurement interval	83631779
Ramp down time	5 minutes

Checkpointing

Number of checkpoints in measurement interval	4
Checkpoint Interval	30 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	20
4.3 60 Day Space	21
4.3.1 Transaction Log Space Requirements.....	21
4.4 Type of Database Used	21
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.....	32
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.....	113
B.1	Database Options.....	144
B.2	Table definitions	144
B.3	Stored Procedures	147
Appendix C	Tunable Parameters.....	200
C.1	Microsoft SQL Server 8.0 Configuration Parameters	240
C.2	Client System Configuration Parameters	240
C.3	RTE Input Parameters	241
Appendix D	60 Day Space Requirements	269
Appendix E	3 rd Party Pricing.....	270

Preface

Document Structure

This is the full disclosure report for a benchmark test of the HP Server rx8620 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 Server rx8620. 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, version 8.00.883.

The processor architecture of the HP Server rx8620 was designed for the Intel Itanium 2 6M processor. The HP Server rx8620 used in this test was powered by 16 1.5GHz Intel Itanium 2 6M processors, each with 6MB of 3rd level cache.

This configuration used 128 GB of HP SDRAM. This was achieved by using 64 2GB 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 mirrored HP MSA1000 diskarrays. Each MSA1000 held 6 146MB Ultra320 SCSI hard drives. The MSA1000 disk arrays were connected to the rx8620 using 2 Emulex 9802 Fibrechannel HBA's. The disks were configured as RAID 0, the disk array controller cache was enabled. The 2 arrays were then configured as a dynamic mirrored disks using the disk mirroring feature of Windows Server 2003 Logical Disk Manager.

The TPC-C database storage consisted of 896 HP18GB 15krpm hard drives. 64 HP StorageWorks 4414R disk arrays were used to connect the disks. Each 4414R holds 14 disks. The 4414R enclosures were attached in groups of 4 to each SmartArray 5304 Controller using standard SCSI cabling. The discs were configured as a RAID0 Array spanning all 56 discs in each array, for a total of 16 RAID0 data base arrays. The SmartArray caches were enabled 50%Read/50% Write on the RAID0 Partitions. Additionally, a RAID0+1 volume was configured on each set of 56 discs as fault tolerant backup for the data base.

Each of the 16 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. 36 remote terminal emulators (RTEs) emulated 241,920 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.

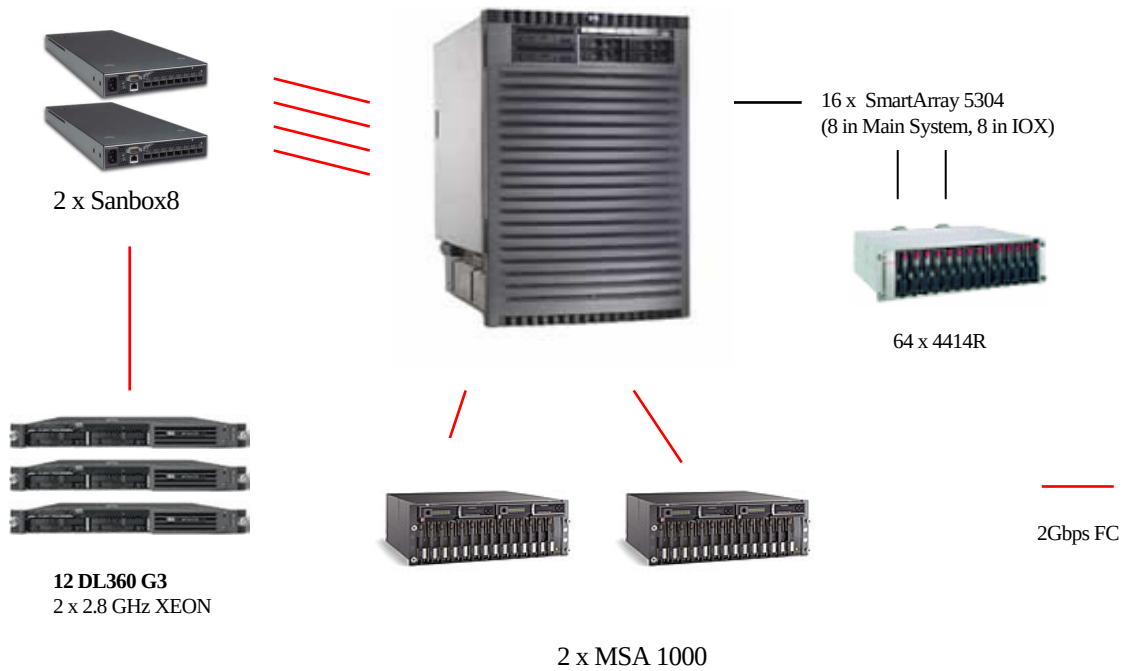
Appendix D contains the 60 day space calculations.

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 and enclosures, Ultra320 SCSI disk and enclosures 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 910 disks, which included 896 18GB Hot Swap disk drives for data, 12 146GB drives for log, and two 18GB drive for the operating system.

Part of the space on each of the 16 database disk arrays was configured as 1 RAID0 volume over 56 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. The remainder of the disc space on each of the 16 Arrays was configured as a RAID0+1 volume over all 56 18GB drives. Each volume had 2 partitions, containing backups of the data base.

1.3 Insert and Delete Operations

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

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

1.4 Partitioning

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

Partitioning was not used on any table.

1.5 Replication, Duplication or Additions

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

No replications, duplications or additional attributes were used.

Chapter 2 Transaction and Terminal Profiles

2.1 Random Number Generation

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

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

2.2 Input/Output Screen Layout

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

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

2.3 Priced Terminal Feature Verification

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

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

2.4 Transaction Statistics

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

Table 1 shows the transaction statistics.

Table 1. Transaction Statistics

Type	Item	Value
New Order	Home warehouse items	99.00%
	Remote warehouse items	1.00%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse	85.02%
	Remote warehouse	14.98%
	Non primary key access	60.01%
Order Status	Non primary key access	60.11%
Delivery	Skipped transactions	0
Transaction Mix	New Order	44.96%
	Payment	43.01%
	Delivery	4.01%
	Stock Level	4.01%
	Order Status	4.01%

2.5 Presentation Manager or Intelligent Terminal

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

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

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

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

2.6 Queuing Mechanism

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

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

The source code is listed in Appendix A.

Chapter 3 Transaction and System Properties

3.1 Transaction System Properties (ACID Tests)

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

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

All of the specified ACID tests were successfully performed on the HP Server rx8620. A fully scaled database was used for all the durability tests.

3.2 Atomicity Tests

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

3.2.1 COMMIT Transaction

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

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

3.2.2 ROLLBACK Transaction

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

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

3.3 Consistency Tests

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

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

3.4 Isolation Tests

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

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

We ran a total of nine isolation tests. Seven of these tests are detailed in the TPC-C specification (clause 3.4.2.1 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. All test were performed on the full scale database..

3.5.1 Loss of Data

The standard driving mechanism was used to generate the transaction load of slightly more than 24,192 users for the test (10% of full load). 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. Slightly more than 24,192 (10%) users were logged in to the database and ran transactions.
4. After 5 minutes, one data disk drive was removed. Errors were noted on both the SQL log, OS log, and the RTE log.
5. The RTE monitor was used to verify that no users were lost.
6. The RTE was shutdown and a success file was created.
7. The database log was backed up to disc.
8. SQL was shut down, the disc re-inserted and the RAID0 volume recovered.
9. The database was restored from the original backup that was restored before the run, specifying recovery NOT be done after the restore.
10. The log was restored with recovery, effectively rolling forward all successful transactions from the run.
11. Transaction were exported from the success file. 6 New Orders were chosen at random and verified to exist in the database.

3.5.2 Loss of System / Memory and loss of Log

This was demonstrated on the full database with 24,192 warehouses in a single test. The standard driving mechanism was used to generate the transaction load of 241,920 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. 241,920 users were logged in to the database and ran transactions.
4. After 5 minutes, one of the (mirrored) log disk was removed from the system, processing transactions continued.
5. After another 5 minutes, power to the system was cut off.
6. The RTE continued running and completed transactions enroute from the clients were recorded. Error messages began appearing on the RTE screen.
7. The RTE was stopped.
8. The server machine was powered on again and rebooted.
9. Microsoft SQL Server was restarted and performed an automatic recovery.
10. A new count of D_NEXT_O_ID was taken.
11. This number was compared with the number of new orders reported by the RTE
12. Samples were taken of the RTE log and verified against the database.

Chapter 4 Scaling and Database Population

4.1 Database Layout

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

The measured (tested) and priced systems used 16 HP Smart Array 5304 Disk Arrays. These cards were plugged into 8 PCI-X slots of the main system, and 8 PCI-X slots on the IO Extender (IOX). The system drive was attached to a Ultra3 LVD SCSI controller, a standard part of the Maintenance Processor shipped with the base system.. The MSA1000 disk arrays that held the database log were connected to 2 Emulex 9802 FC HBA's which were plugged into 2 PCI-X slots in the main system chassis.

The measured database configuration used a total of 910 disks, which included 896 18GB Hot Swap disk drives for data, 12 146GB drives for the log, and two 18GB drive for the operating system. Part of the space on each of the 16 database disk arrays was configured as 1 RAID0 volume over 56 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. The remainder of the disc space on each of the 16 Arrays was configured as a RAID0+1 volume over all 56 18GB drives. Each volume had 2 partitions, containaining backups of the data base.

Table 2 shows the complete data distribution.

Table 2: Data Distribution

SmartArray 5304 HBA #1					WINDOWS.NET DISK ADMIN	
4 StorageWorks					DISK 1	
SCSI ID	Channels				Partitions (RAID 0):149GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\CS1 Raw 96GB	G:\MNT\MISC1 Raw 53GB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
8	18GB	18GB	18GB	18GB		
9	18GB	18GB	18GB	18GB		
10	18GB	18GB	18GB	18GB		
11	18GB	18GB	18GB	18GB		
12	18GB	18GB	18GB	18GB		
13	18GB	18GB	18GB	18GB		
14	18GB	18GB	18GB	18GB		
SmartArray 5304 HBA #1					WINDOWS.NET DISK ADMIN	
4 StorageWorks					DISK 2	
SCSI ID	Channels				Partitions (RAID 0+1):400GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\BACKUP\1\ NTFS Volume Backup1 134GB	G:\MNT\BACKUP2\1\ NTFS Volume \Backup1 134GB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
8	18GB	18GB	18GB	18GB		
9	18GB	18GB	18GB	18GB		
10	18GB	18GB	18GB	18GB		
11	18GB	18GB	18GB	18GB		
12	18GB	18GB	18GB	18GB		
13	18GB	18GB	18GB	18GB		
14	18GB	18GB	18GB	18GB		

SmartArray 5304 HBA #2					WINDOWS.NET DISK ADMIN	
4 StorageWorks					DISK 3	
SCSI ID	Channels				Partitions (RAID 0):149GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\CS2 Raw 96GB	G:\MNT\MISC2 Raw 53GB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
8	18GB	18GB	18GB	18GB		
9	18GB	18GB	18GB	18GB		
10	18GB	18GB	18GB	18GB		
11	18GB	18GB	18GB	18GB		
12	18GB	18GB	18GB	18GB		
13	18GB	18GB	18GB	18GB		
14	18GB	18GB	18GB	18GB		
SmartArray 5304 HBA #2					WINDOWS.NET DISK ADMIN	
4 StorageWorks					DISK 4	
SCSI ID	Channels				Partitions (RAID 0+1):400GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\BACKUP2\ NTFS Volume Backup2 134GB	G:\MNT\BACKUP2\ NTFS Volume Backup2 134GB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
8	18GB	18GB	18GB	18GB		
9	18GB	18GB	18GB	18GB		
10	18GB	18GB	18GB	18GB		
11	18GB	18GB	18GB	18GB		
12	18GB	18GB	18GB	18GB		
13	18GB	18GB	18GB	18GB		
14	18GB	18GB	18GB	18GB		

....

SmartArray 5304 HBA #16					WINDOWS.NET DISK ADMIN	
4 StorageWorks					DISK 31	
SCSI ID	Channels				Partitions (RAID 0):149GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\CS16 Raw 96GB	G:\MNT\MISC16 Raw 53GB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
8	18GB	18GB	18GB	18GB		
9	18GB	18GB	18GB	18GB		
10	18GB	18GB	18GB	18GB		
11	18GB	18GB	18GB	18GB		
12	18GB	18GB	18GB	18GB		
13	18GB	18GB	18GB	18GB		
14	18GB	18GB	18GB	18GB		
SmartArray 5304 HBA #16					WINDOWS.NET DISK ADMIN	
4 StorageWorks					DISK 32	
SCSI ID	Channels				Partitions (RAID 0+1):400GB	
	0	1	2	3	0	1
0	18GB	18GB	18GB	18GB	G:\MNT\BACKUP\16\ NTFS Volume Backup16 134GB	G:\MNT\BACKUP2\16\ NTFS Volume Backup16 134GB
1	18GB	18GB	18GB	18GB		
2	18GB	18GB	18GB	18GB		
3	18GB	18GB	18GB	18GB		
4	18GB	18GB	18GB	18GB		
5	18GB	18GB	18GB	18GB		
6	18GB	18GB	18GB	18GB		
8	18GB	18GB	18GB	18GB		
9	18GB	18GB	18GB	18GB		
10	18GB	18GB	18GB	18GB		
11	18GB	18GB	18GB	18GB		
12	18GB	18GB	18GB	18GB		
13	18GB	18GB	18GB	18GB		
14	18GB	18GB	18GB	18GB		

4.2 Initial Cardinality of Tables

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

Table 3 shows the cardinality of the various tables.

Table 3: Table Cardinality

Table	Occurrences
Warehouse	26,000
District	260,000
Customer	780,000,000
History	780,000,000
Orders	780,000,000
New Orders	234,000,000
Order Line	7,800,014,536
Stock	2,600,000,000
Item	100,000

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 646 GB to sustain the log for 8 hours. Space available for the transaction log was 820 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_fg and cs_fg. misc_fg consists of 16 partitions at 96,568 MB each and misc_fg consist of 16 partitions at 53,520 MB each as shown in the createdb.sql. The log was configured with 20,000 MB at database creation, and was expanded to 300GB after data base creation and load.

Chapter 5 Performance Metrics and Response Time

5.1 Throughput

Measured tpmC® must be reported.

Measured TpmC®: 301,225.52
Price per TpmC®: \$4.56

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	0.54s	0.90s	17.78s
Payment	0.48s	0.84s	56.09s
Order-Status	0.52s	0.87s	13.57s
Delivery (interactive portion)	0.10s	0.11s	1.20s
Delivery (deferred portion)	0.14s	0.19s	3.88s
Stock-Level	1.08s	1.59s	16.74s
Menu	0.10s	0.11s	6.46s

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

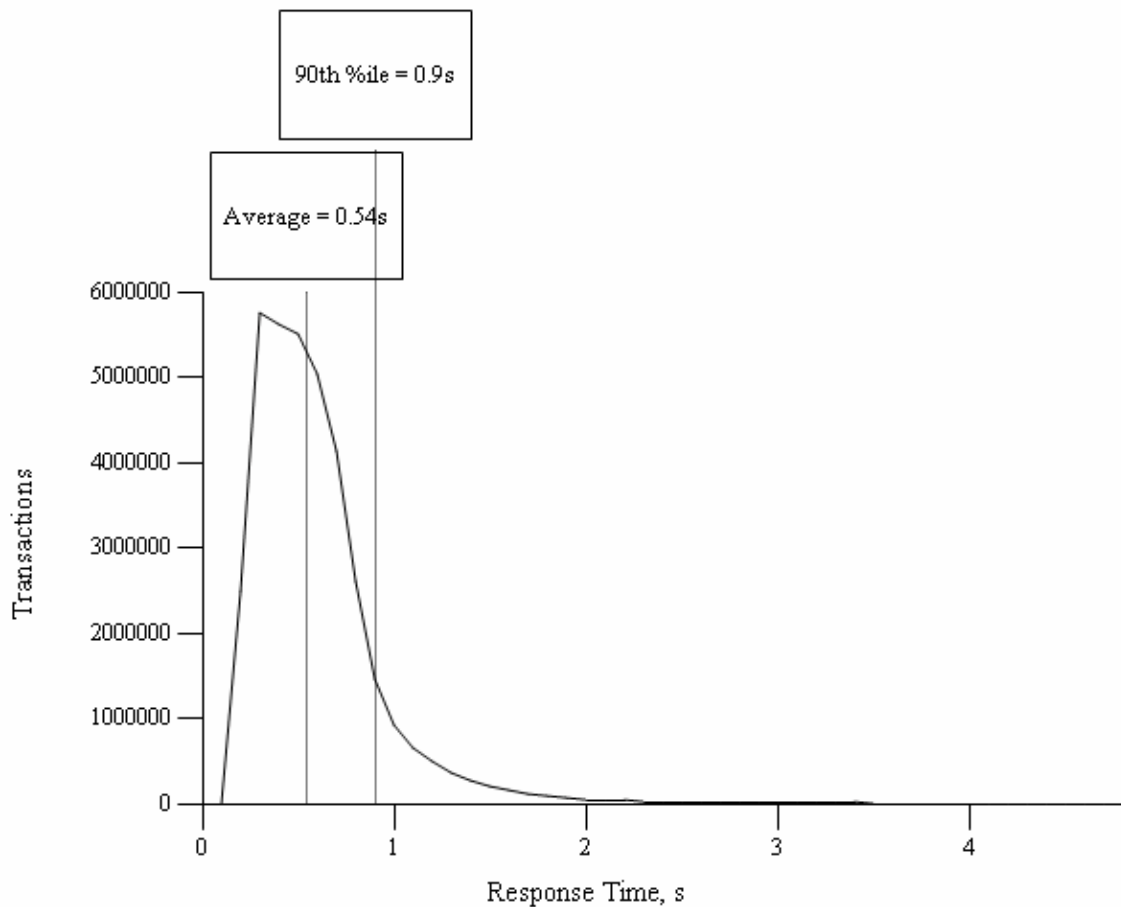
Think Times	Minimum	Average	Maximum
New Order	0s	12.05s	120.51s
Payment	0s	12.04s	120.51s
Order Status	0s	10.05s	100.5s
Interactive Delivery	0s	5.05s	50.51s
Stock Level	0s	5.06s	50.5s

5.4 Response Time Frequency

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

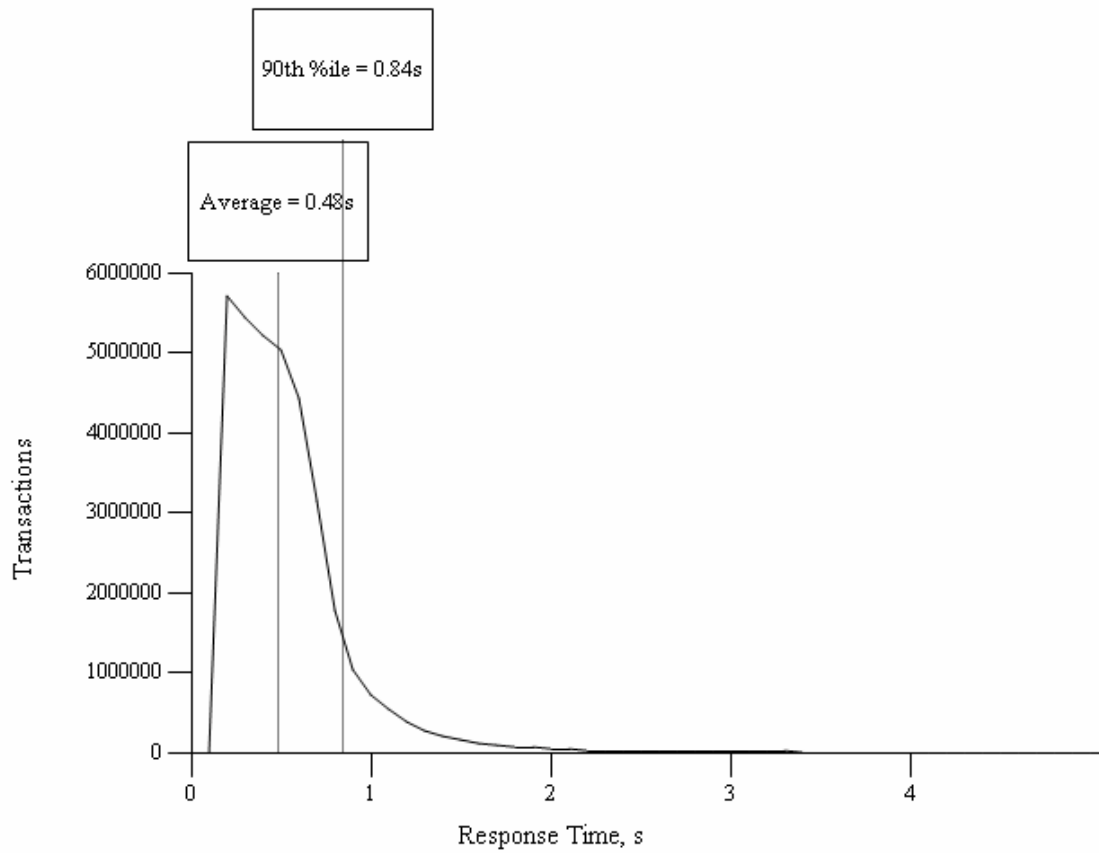
Figure 3: New Order Response Time Distribution



Response time frequency distribution for New Order transaction

5.4.2 Payment Response Time Distribution

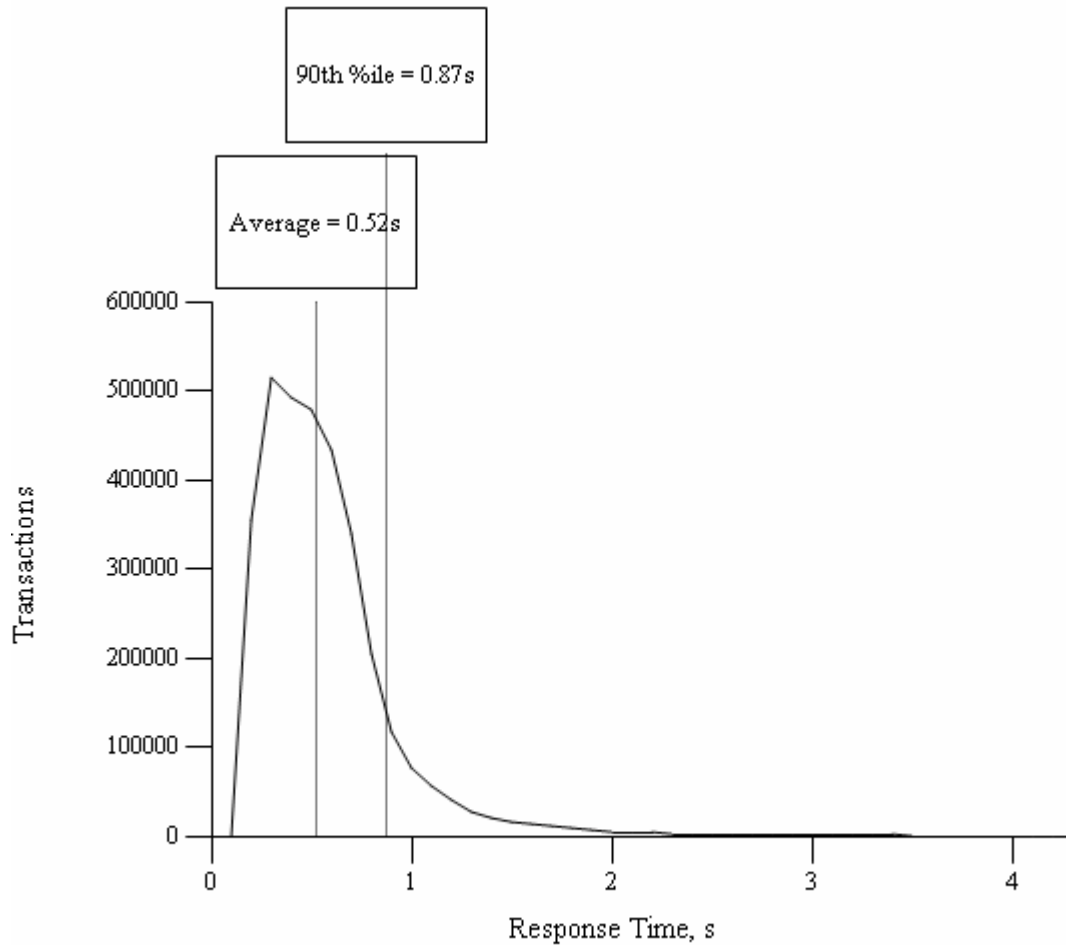
Figure 4: Payment Response Time Distribution



Response time frequency distribution for Payment transaction

5.4.3 Order Status Response Time

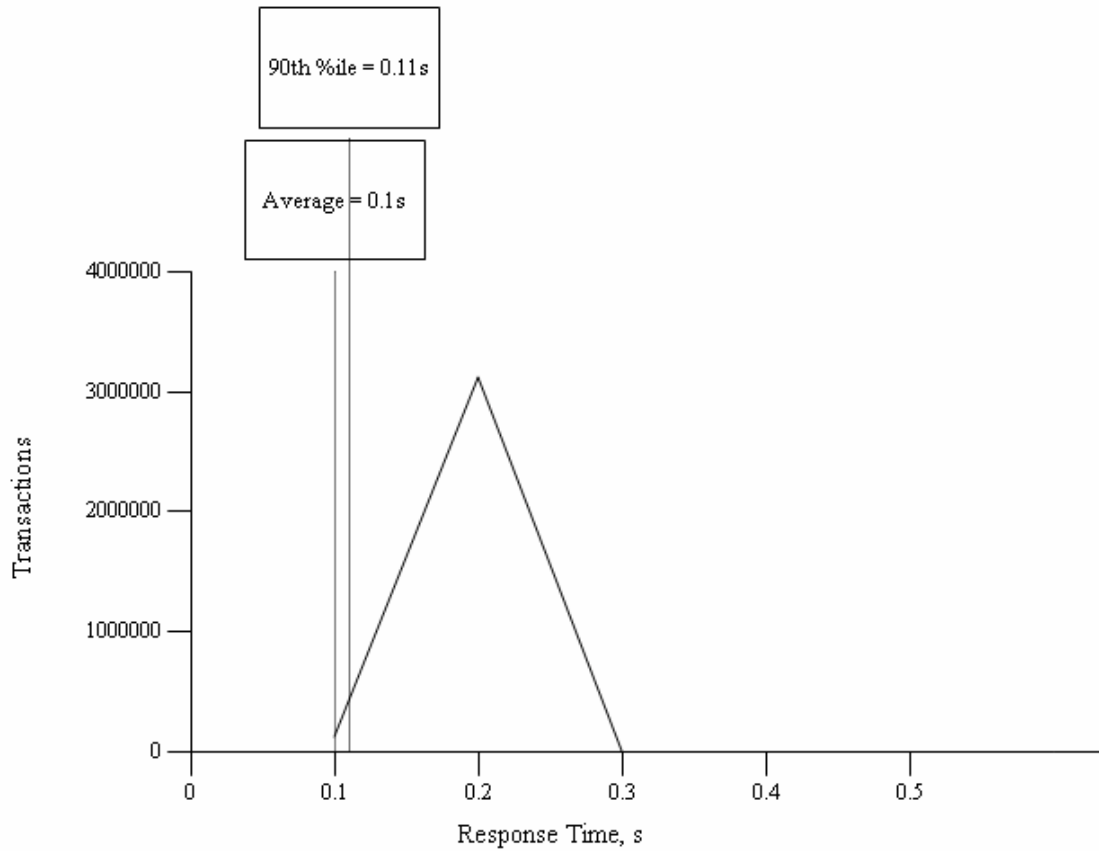
Figure 5: Order Status Response Time Distribution



Response time frequency distribution for Order Status transaction

5.4.4 Delivery Response Time Distribution

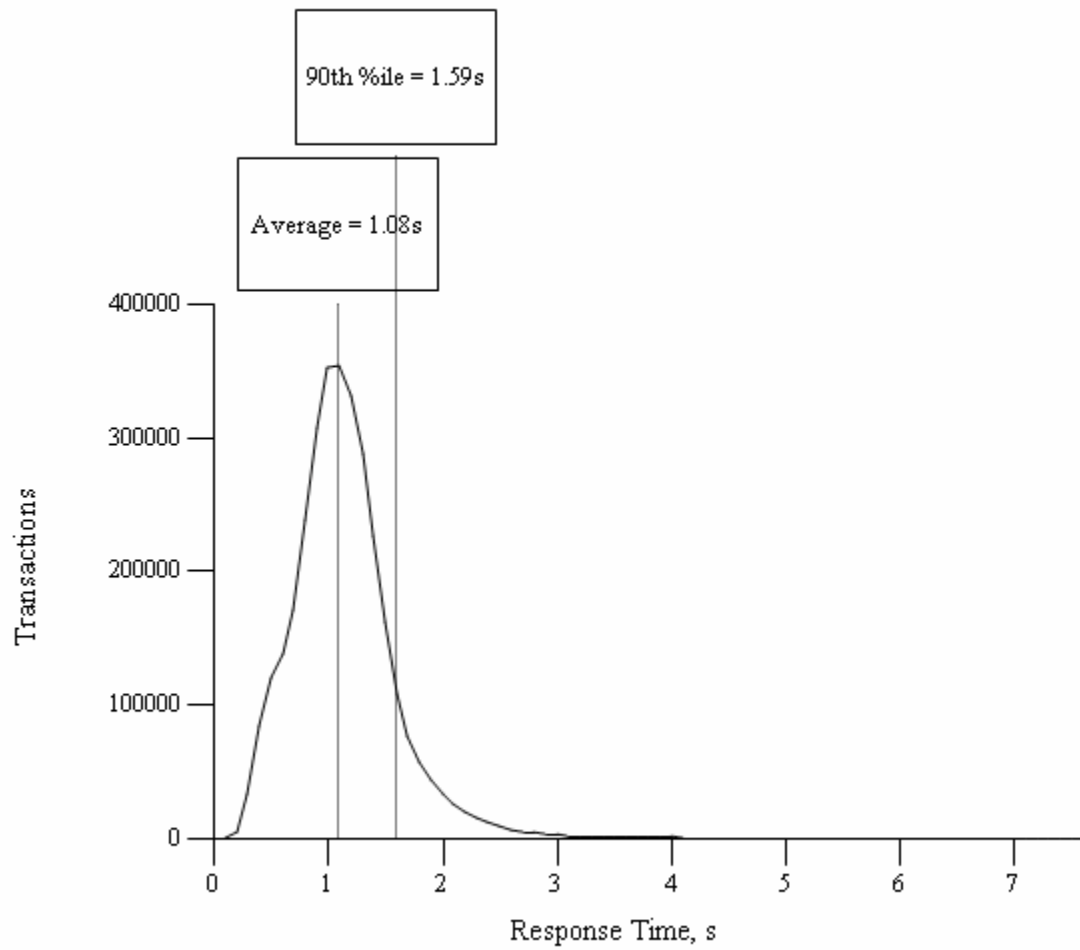
Figure 6: Delivery Response Time Distribution



Response time frequency distribution for Delivery transaction

5.4.5 Stock Level Response Time

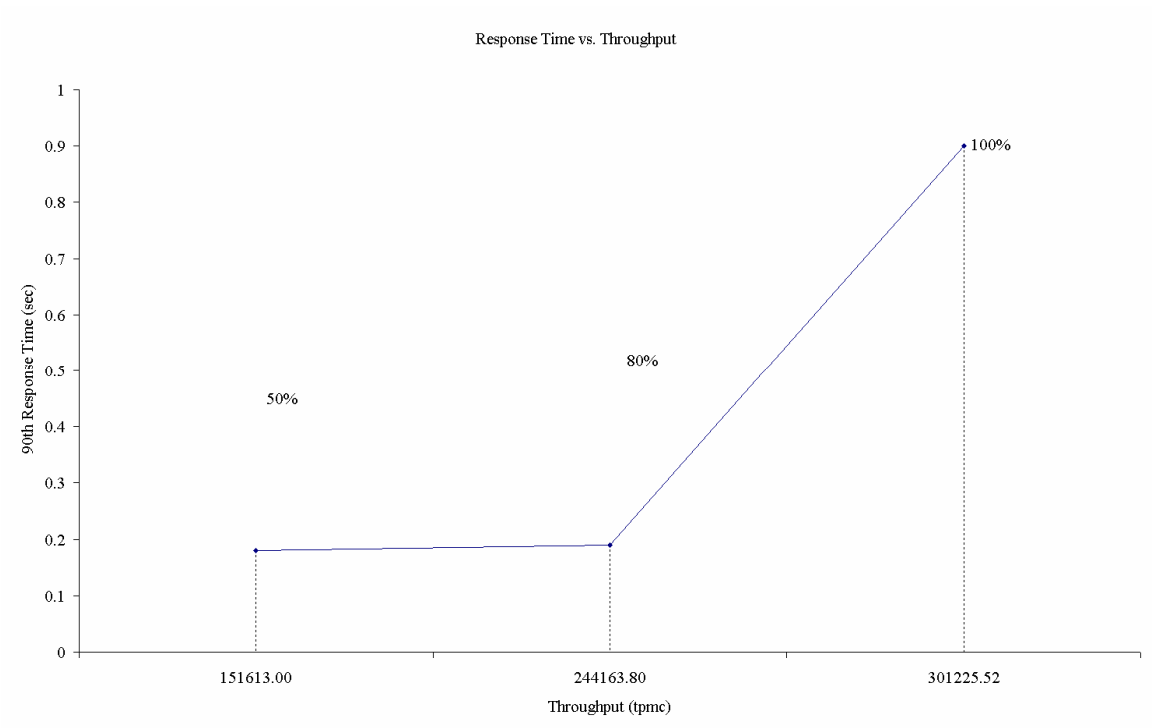
Figure 7: Stock Level Response Time Distribution



Response time frequency distribution for Stock Level transaction

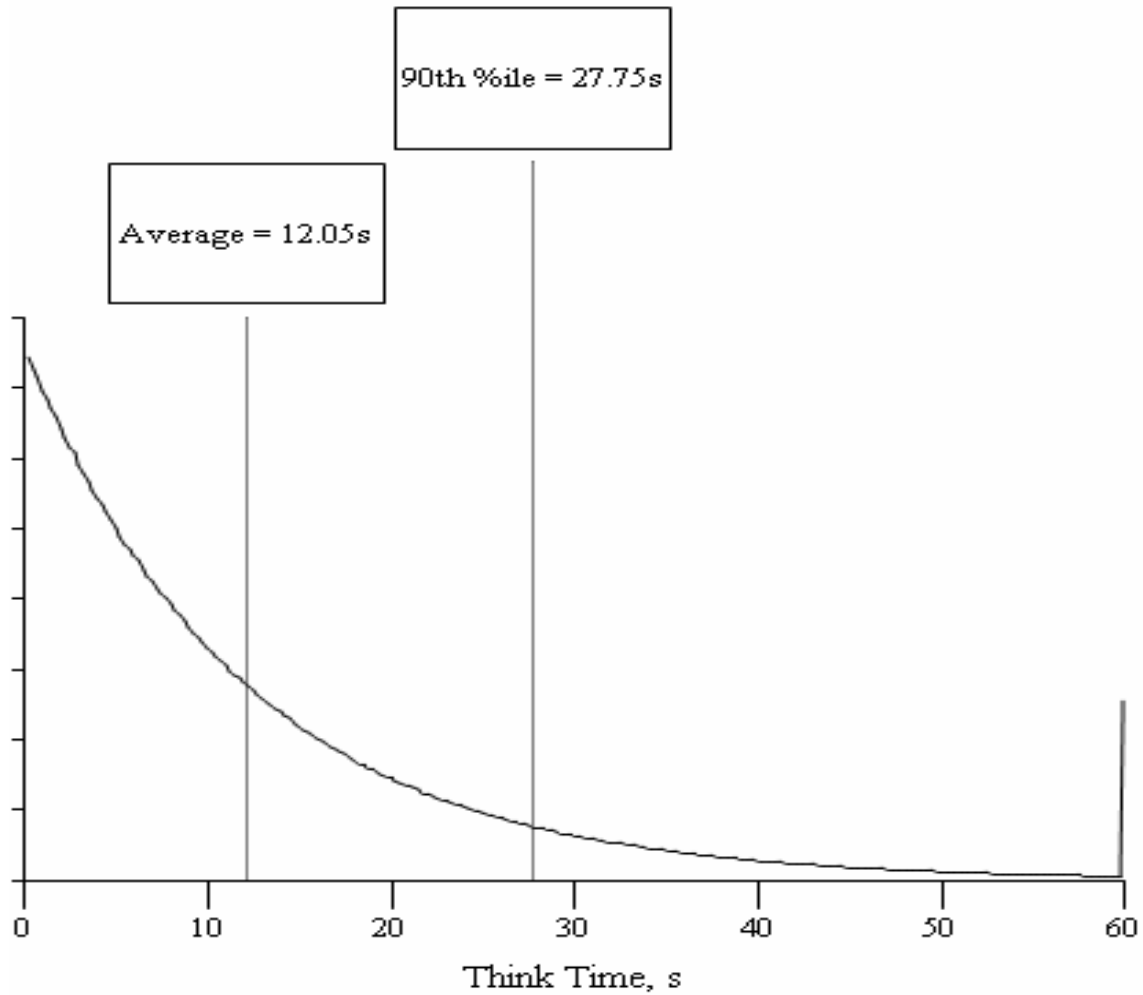
5.4.6 Response Time Versus Throughput

Figure 8: New Order Response Time Distribution



5.4.7 New Order Think Time Distribution

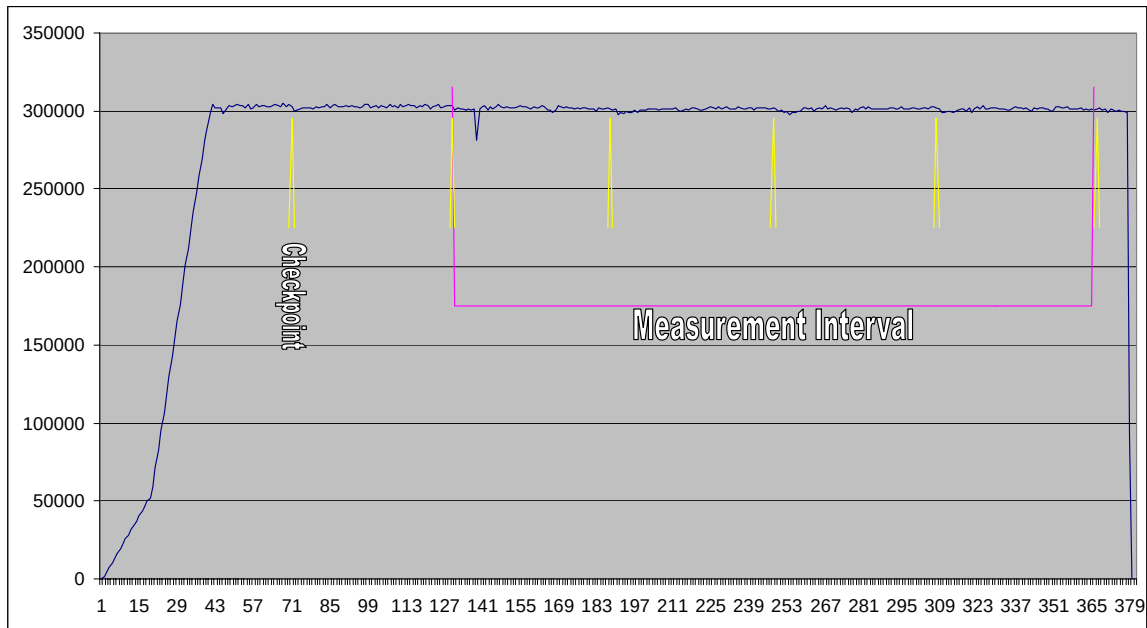
Figure 9: New Order Think Time Distribution



Think time frequency distribution for New Order transaction

5.4.8 Throughput Versus Time Distribution

Figure 10: New Order Throughput versus Time



5.5 Steady State Determination

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

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

5.6 Work Performed During Steady State

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

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

5.6.1 Checkpoint

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

5.6.2 Checkpoint Conditions

There are two types of checkpoints:

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

Forcing dirty pages onto the database device means that all completed transactions are written out. By 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. SQL was configured with the sp_configure “max recovery interval”,119 command to issue checkpoints of 29 minutes, 45 seconds. The checkpoints were affinitized to cell 3 using connection affinity. One of the web clients whose network connections were affinitized to that cell was used to issue the checkpoints. This was done with the same QL2350 VI card that was used for network traffic. 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	14:04:02	16:04:02
Checkpoint	13:34:16	14:04:02
Checkpoint	14:04:02	14:33:47
Checkpoint	14:33:47	15:03:33
Checkpoint	15:03:33	15:33:18
Checkpoint	15:33:18	16:03:04
Checkpoint	16:03:04	16:25:02

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

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

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

7.2 General Availability, Throughput and Price Performance

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

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

Table 9: Throughput, Price Performance and Availability

Maximum qualified throughput:	301225.52 tpmC
Price per tpmC:	\$4.56
Availability:	April 15, 2004

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:

- 16 Microsoft SQL Server 2000 Enterprise Edition 64-bit per-processor licenses.
- Microsoft Windows Server 2003, Datacenter edition (64-bit)
- 12 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:

December 10, 2003

MR. JASON GOERTZ
PERFORMANCE ENGINEER
WIE Redmond
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 rx8620 16P
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 rx8620 Server with:				
CPU's	Memory	Disks (total)	90% Response	TpmC
16 Itanium 2 @ 1.5 Ghz	Main: 138 GB	12 @ 146 GB 896 @ 18 GB 2 @ 18 GB (os)	0.90	301,225.52
12 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 26,000 warehouses.
- The ACID properties were successfully demonstrated.
- The durability data loss and log loss tests were demonstrated the full database.
- 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.
- The Measured cycle times were confirmed to have the correct response time delays.
- There were 260,000 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.
- Four checkpoints were contained completely inside the measured interval.

- Checkpoint interval was verified to be less than 30 minutes.
- The system pricing was checked for major components and maintenance.
- Third party quotes were verified for compliance.

Auditor Notes:

The measured configuration had 4 full arrays populated with 56 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.

Sincerely,

A handwritten signature in cursive script that reads "Lorna Livingtree".

Lorna Livingtree
Auditor

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
 *
 *      Kit Ver. 4.20.000
 *
 *      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 support

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 NotIntErr);
void TermInit(void);
void TermDeleteAll(void);
int TermAdd(void);
void TermDelete(int id);
void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int
iErrorNum, int iTermId, int iSyncId, char *szErrorText, char *szBuffer );
void MakeMainMenuForm(int iTermId, int iSyncId, char *szForm);
void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA
*pStockLevelData, BOOL bInput, char *szForm);
void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA
*pNewOrderData, BOOL bInput, char *szForm);
void MakePaymentForm(int iTermId, PAYMENT_DATA *pPaymentData,
BOOL bInput, char *szForm);
void MakeOrderStatusForm(int iTermId, ORDER_STATUS_DATA
*pOrderStatusData, BOOL bInput, char *szForm);
void MakeDeliveryForm(int iTermId, DELIVERY_DATA *pDeliveryData,
BOOL bInput, char *szForm);
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId, char *szBuffer);
void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA
*pNewOrderData);
void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA
*pPaymentData);
void GetOrderStatusData(LPSTR lpszQueryString,
ORDER_STATUS_DATA *pOrderStatusData);
BOOL PostDeliveryInfo(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)
#ifndef _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 "OriginalName", "Microsoft\0"
        VALUE "ProductVersion", "0, 4, 0, 0\0"
    END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
    END
#endif // !_MAC

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

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

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

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

#endif // APSTUDIO_INVOKED

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

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

```
////////////////////
//
// DESIGNINFO
//

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

#ifdef // 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

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

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

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

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

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

```

```

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

            DisableThreadLibraryCalls((HMODULE)hModule);
            InitializeCriticalSection(&TermCriticalSection);

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

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

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

                TermInit();

                // load DLL for
                if (Reg.eTxnMon
                    == TUXEDO)
                {
                    strcpy( szDllName, Reg.szPath );
                    strcat( szDllName, "tpcc_tuxedo.dll");

                    hLibInstanceTm = LoadLibrary( szDllName );

                    if
                    (hLibInstanceTm == NULL)
                    {
                        throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED,
                            szDllName, GetLastError() );
                    }
                }
            }
        }
    }

```

```

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

```

```

//
get function pointer to wrapper for class constructor
pCTPCC_COM_new = (TYPE_CTPCC_COM*)
GetProcAddress(hLibInstanceTm,"CTPCC_COM_new");
if
(pCTPCC_COM_new == NULL)
{
    throw new CWEBCLNT_ERR(
        ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
    }
    // load DLL for
    database connection
    if ((Reg.eTxnMon
        == None) || (dwNumDeliveryThreads > 0))
    {
        if
        (Reg.eDB_Protocol == DBLIB)
        {
            strcpy( szDllName, Reg.szPath );
            strcat( szDllName, "tpcc_dblib.dll");
            hLibInstanceDb = LoadLibrary( szDllName );
            if (hLibInstanceDb == NULL)
            {
                throw new CWEBCLNT_ERR(
                    ERR_LOADDLL_FAILED, szDllName, GetLastError() );
            }
            // get function pointer to wrapper for class constructor
            pCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*)
            GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_new");
            if (pCTPCC_DBLIB_new == NULL)
            {
                throw new CWEBCLNT_ERR(
                    ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
            }
            }
            else
            if (Reg.eDB_Protocol == ODBC)
            {
                strcpy( szDllName, Reg.szPath );
                strcat( szDllName, "tpcc_odbc.dll");
                hLibInstanceDb = LoadLibrary( szDllName );
            }
        }
    }

```

```

        if (hLibInstanceDb == NULL)

            throw new CWEBCLNT_ERR(
ERR_LOADDLL_FAILED, szDllName, GetLastError() );

        // get function pointer to wrapper for class constructor

        pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");

        if (pCTPCC_ODBC_new == NULL)

            throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
    }

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

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

            InitializeCriticalSection(&DelBuffCriticalSection);

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

            dwDelBuffFreeCount = dwDelBuffSize;

            InitJulianTime(NULL);

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

            SYSTEMTIME Time;

            GetLocalTime( &Time );

            wsprintf( szLogFile, "%sdelivery-%2.2d%2.2d%2.2d-
            %2.2d%2.2d.log",

                Reg.szPath, Time.wYear % 100,
                Time.wMonth, Time.wDay, Time.wHour, Time.wMinute );

            txnDelilog = new CTxnLog(szLogFile, TXN_LOG_WRITE);

```

```

        //write event into txn log for START

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

        //
        allocate structures for delivery buffers and thread mgmt

        pDeliHandles = new HANDLE[dwNumDeliveryThreads];

        pDelBuff = new
DELIVERY_TRANSACTION[dwDelBuffSize];

        //
        launch DeliveryWorkerThread to perform actual delivery txns

        for(i=0; i<dwNumDeliveryThreads; i++)
        {
            pDeliHandles[i] = (HANDLE) _beginthread(
DeliveryWorkerThread, 0, NULL );

            if (pDeliHandles[i] == INVALID_HANDLE_VALUE)

                throw new CWEBCLNT_ERR(
ERR_DELIVERY_THREAD_FAILED );
        }

        break;

        case DLL_PROCESS_DETACH:
            if
(dwNumDeliveryThreads)
            {
                if
(txnDelilog != NULL)

                    //write event into txn log for STOP

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

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

                    CTxnLog *txnDelilogLocal = txnDelilog;

                    txnDelilog= NULL;

                    delete txnDelilogLocal;

```

```

    }

    delete [] pDeliHandles;

    delete [] pDelBuff;

    CloseHandle( hWorkerSemaphore );

    CloseHandle( hDoneEvent );

    DeleteCriticalSection(&DelBuffCriticalSection);
}

DeleteCriticalSection(&TermCriticalSection);

if
(hLibInstanceTm != NULL)

    FreeLibrary( hLibInstanceTm );

    hLibInstanceTm
= NULL;

if
(hLibInstanceDb != NULL)

    FreeLibrary( hLibInstanceDb );

    hLibInstanceDb =
NULL;

    Sleep(500);
    break;

    default:
        /* nothing */;
}

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

```

```

        return TRUE;
    }

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

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

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

    return TRUE;
}

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

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

```

```

    }
    TermDeleteAll();
    return TRUE;
}

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

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

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

#ifdef ICECAP
    StartCAP();
#endif

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

```

```

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

                WriteMessageToEventLog( szTmp );

                throw new
                CWEBCLNT_ERR( ERR_INVALID_TERMID );
            }

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

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

        switch(iCmd)
        {
        case 0:
            WelcomeForm(pECB, szBuffer);
            break;

        case 1:
            switch( FormId )
            {
            case
            WELCOME_FORM:
            case
            MAIN_MENU_FORM:
                break;

            case
            NEW_ORDER_FORM:
                ProcessNewOrderForm(pECB, TermId, szBuffer);
                break;

            case
            PAYMENT_FORM:
                ProcessPaymentForm(pECB, TermId, szBuffer);
            }
        }
    }

```

```

        break;

        case
DELIVERY_FORM:
        ProcessDeliveryForm(pECB, TermId, szBuffer);

        break;

        case
ORDER_STATUS_FORM:
        ProcessOrderStatusForm(pECB, TermId, szBuffer);

        break;

        case
STOCK_LEVEL_FORM:
        ProcessStockLevelForm(pECB, TermId, szBuffer);

        break;
    }
    break;

    case 2:
        // new-order selected from
        menu; display new-order input form
        NULL, INPUT_FORM, szBuffer);
        break;

    case 3:
        // payment selected from
        menu; display payment input form
        NULL, INPUT_FORM, szBuffer);
        break;

    case 4:
        // delivery selected from menu;
        display delivery input form
        NULL, INPUT_FORM, szBuffer);
        break;

    case 5:
        // order-status selected from
        menu; display order-status input form
        NULL, INPUT_FORM, szBuffer);
        break;

    case 6:
        // stock-level selected from
        menu; display stock-level input form
        NULL, INPUT_FORM, szBuffer);
        break;

```

```

        case 7:    // ExitCmd
                   TermDelete(TermId);
                   WelcomeForm(pECB, szBuffer);
                   break;

        case 8:
                   SubmitCmd(pECB, szBuffer);
                   break;

        case 9:
                   // menu
                   MakeMainMenuForm(TermId,
Term.pClientData[TermId].iSyncId, szBuffer);
                   break;

        case 10:
                   // CMD=Clear
                   // resets all connections;
                   should only be used when no other connections are active
                   TermDeleteAll();
                   TermInit();
                   WelcomeForm(pECB, szBuffer);
                   break;

        case 11:
                   // CMD=Stats
                   StatsCmd(pECB, szBuffer);
                   break;
    }

    catch (CBaseErr *e)
    {
        ErrorForm( pECB, e->ErrorType(), e-
>ErrorNum(), TermId, iSyncId, e->ErrorText(), szBuffer );
        delete e;
    }

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

#ifdef ICECAP
    StopCAP();
#endif

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

```

```

        (*pECB->ServerSupportFunction)(pECB->ConnID,
HSE_REQ_SEND_RESPONSE_HEADER, szHeader, (LPDWORD) &dwSize,
(LPDWORD)szHeader1);
        //finish up and keep connection
        pECB->dwHttpStatusCode = 200;
        return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
    }

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

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

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

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

        (VOID) DeregisterEventSource(hEventSource);
    }
}

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

```

```

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

    DELIVERY_TRANSACTION delivery;
    PDELIVERY_DATA pDeliveryData;
    TXN_RECORD_TPCC_DELIV_DEF txnDeliRec;

    DWORD index;
    HANDLE handles[2];

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

    assert(txnDeliRec != NULL);

    try
    {
        if (Reg.eDB_Protocol == ODBC)
            pTxn = pCTPCC_ODBC_new(
                Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
                Reg.szMyComputerName, Reg.szDbName );
        else if (Reg.eDB_Protocol == DBLIB)
            pTxn = pCTPCC_DBLIB_new(
                Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
                Reg.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,
        //need to wait
        while (TRUE)
        {
            // need to wait
            for multiple objects: program exit or worker semaphore;
            handles[0] =
            hDoneEvent;
            handles[1] =
            hWorkerSemaphore;
            index =
            WaitForMultipleObjects( 2, &handles[0], FALSE, INFINITE );
            if (index ==
            WAIT_OBJECT_0)
                goto
            ErrorExit;

            ZeroMemory(&txnDeliRec, sizeof(txnDeliRec));

            txnDeliRec.TxnType = TXN_REC_TYPE_TPCC_DELIV_DEF;

            // make a local
            copy of current entry from delivery buffer and increment buffer index

            EnterCriticalSection(&DelBuffCriticalSection);
            delivery =
            *(pDelBuff+dwDelBuffBusyIndex);

            dwDelBuffFreeCount++;

            dwDelBuffBusyIndex++;

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

            dwDelBuffBusyIndex = 0;

            LeaveCriticalSection(&DelBuffCriticalSection);

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

```

```

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

            txnDeliRec.TxnStartT0 = Get64BitTime(&delivery.queue);

            GetLocalTime(
            &trans_start );

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

            //log txn

            txnDeliRec.TxnStatus = ERR_SUCCESS;
            for (int i=0; i<10;
            i++)

            txnDeliRec.o_id[i] = pDeliveryData->o_id[i];

            txnDeliRec.DeltaT4 = (int)(Get64BitTime(&trans_end) -
            txnDeliRec.TxnStartT0);

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

            if (txnDeliRec !=
            NULL)

            txnDeliRec->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 (txnDeliRec != NULL)
                txnDeliRec->
                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;

        GetLocalTime(&(pDelBuff+dwDelBuffFreeIndex)->queue);

        dwDelBuffFreeCount--;
        dwDelBuffFreeIndex++;
        if (dwDelBuffFreeIndex == dwDelBuffSize)
            dwDelBuffFreeIndex = 0;

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

    LeaveCriticalSection(&DelBuffCriticalSection);

```

```

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

        return bError;
    }

/* FUNCTION: ProcessQueryString
*
* PURPOSE: This function extracts the 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..", "..Stock-Level..",
"..Exit..", "Submit", "Menu", "Clear", "Stats",
""
};

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

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

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

```

```

        *pSyncId = GetIntKeyValue(&ptr, "SYNCID", NO_ERR,
NO_ERR);
    // parse CMD
    GetIntKeyValue(&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=\"\"><BR>"

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

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

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

"</PRE></font>"

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

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

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

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

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

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

"</PRE></font>"

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

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

"<font face=\"Courier New\" color=\"blue\"><PRE>" ;
    strcat( szBuffer, szTmp);
    strcat( szBuffer, "Warehouse ID = <INPUT
NAME=\"w_id\" SIZE=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 (...)
        {
            Term.Delete(iNewTerm);
            throw; // pass exception
        }

        upward

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

/* FUNCTION: StatsCmd
 *
 * PURPOSE: This function returns to the browser the total number of
active terminal ids.

```

```

 *
 * This routine is for
development/debugging purposes.
 */

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

    EnterCriticalSection(&TermCriticalSection);

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

    LeaveCriticalSection(&TermCriticalSection);

    wsprintf( szBuffer,

        "<HTML><HEAD><TITLE>TPC-C Web Client
Stats</TITLE></HEAD>"

        "<BODY><B><BIG> Total
Active Connections: %d </BIG></B><BR></BODY></HTML>"
        , iTotal );
}

char *CWEBCLNT_ERR::ErrorText()
{
    static SERRORMSG errorMsgs[] =
    {
        { ERR_COMMAND_UNDEFINED,

            "Command undefined."

        },

        { ERR_D_ID_INVALID,

            "Invalid District ID Must be 1 to 10."

        },

        {
ERR_DELIVERY_CARRIER_ID_RANGE,
            "Delivery Carrier ID out of range must be 1 - 10."
        },

        {
ERR_DELIVERY_CARRIER_INVALID,
            "Delivery Carrier ID invalid must be numeric 1 - 10."
        },
    }
}

```

```

    {
ERR_DELIVERY_MISSING_OCD_KEY,
        "Delivery missing Carrier ID key \"OCD*\"."
    },
    {
ERR_DELIVERY_THREAD_FAILED,
        "Could not start delivery worker thread."
    },
    {
ERR_GETPROCADDR_FAILED,

        "Could not map proc in DLL. GetProcAddr error. DLL="
        },
    {
ERR_HTML_ILL_FORMED,
        "Required key
field is missing from HTML string."
    },
    {
ERR_INVALID_SYNC_CONNECTION,
        "Invalid Terminal Sync ID."
    },
    {
ERR_INVALID_TERMID,

        "Invalid Terminal ID."
    },
    {
ERR_LOADDLL_FAILED,

        "Load of DLL failed. DLL="
        },
    {
ERR_MAX_CONNECTIONS_EXCEEDED,
        "No connections available. Max Connections is probably
too low."
    },
    {
ERR_MISSING_REGISTRY_ENTRIES,
        "Required registry entries are missing. Rerun INSTALL to
correct."
    },
    {
ERR_NEWORDER_CUSTOMER_INVALID,
        "New Order customer id invalid data type, range = 1 to
3000."
    },
    {
ERR_NEWORDER_CUSTOMER_KEY,
        "New Order missing Customer key \"CID*\"."
    },
}

```

```

    {
      ERR_NEWORDER_DISTRICT_INVALID,
      "New Order District ID Invalid range 1 - 10."
    },
    {
      ERR_NEWORDER_FORM_MISSING_DID,
      "New Order missing District key \"DID*\"."
    },
    {
      ERR_NEWORDER_ITEMID_INVALID,
      "New Order Item Id is wrong data type, must be
numeric."
    },
    {
      ERR_NEWORDER_ITEMID_RANGE,
      "New Order Item Id is out of range. Range =
1 to 999999."
    },
    {
      ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
      "New Order Item_Id field entered without a corresponding
Supp_W."
    },
    {
      ERR_NEWORDER_MISSING_IID_KEY,
      "New Order missing Item Id key \"IID*\"."
    },
    {
      ERR_NEWORDER_MISSING_QTY_KEY,
      "New Order Missing Qty key \"Qty##*\"."
    },
    {
      ERR_NEWORDER_MISSING_SUPPW_KEY,
      "New Order missing Supp_W key
\"SP##*\"."
    },
    {
      ERR_NEWORDER_NOITEMS_ENTERED,
      "New Order No order lines entered."
    },
    {
      ERR_NEWORDER_QTY_INVALID,
      "New Order Qty invalid must be numeric
range 1 - 99."
    },
    {
      ERR_NEWORDER_QTY_RANGE,
      "New Order Qty is out of
range. Range = 1 to 99."
    },
  },

```

```

    {
      ERR_NEWORDER_QTY_WITHOUT_SUPPW,
      "New Order Qty field entered without a
corresponding Supp_W."
    },
    {
      ERR_NEWORDER_SUPPW_INVALID,
      "New Order Supp_W invalid data type must
be numeric."
    },
    {
      ERR_NO_SERVER_SPECIFIED,
      "No Server name
specified."
    },
    {
      ERR_ORDERSTATUS_CID_AND_CLT,
      "Order Status Only Customer ID or Last Name may be
entered, not both."
    },
    {
      ERR_ORDERSTATUS_CID_INVALID,
      "Order Status Customer ID invalid, range must be numeric
1 - 3000."
    },
    {
      ERR_ORDERSTATUS_CLT_RANGE,
      "Order Status Customer last name longer
than 16 characters."
    },
    {
      ERR_ORDERSTATUS_DID_INVALID,
      "Order Status District invalid, value must be numeric 1 -
10."
    },
    {
      ERR_ORDERSTATUS_MISSING_CID_CLT,
      "Order Status Either Customer ID or Last Name must be
entered."
    },
    {
      ERR_ORDERSTATUS_MISSING_CID_KEY,
      "Order Status missing Customer key \"CID*\"."
    },
    {
      ERR_ORDERSTATUS_MISSING_CLT_KEY,
      "Order Status missing Customer Last Name key \"CLT*\"."
    },
    {
      ERR_ORDERSTATUS_MISSING_DID_KEY,
      "Order Status missing District key \"DID*\"."
    },
    {
      ERR_PAYMENT_CDI_INVALID,
      "Payment
Customer district invalid must be numeric."
    },
    {
      ERR_PAYMENT_CID_AND_CLT,
      "Payment Only
Customer ID or Last Name may be entered, not both."
    },
  },

```

```

    {
      ERR_PAYMENT_CUSTOMER_INVALID,
      "Payment Customer data type invalid, must be numeric."
    },
    {
      ERR_PAYMENT_CWI_INVALID,
      "Payment
Customer Warehouse invalid, must be numeric."
    },
    {
      ERR_PAYMENT_DISTRICT_INVALID,
      "Payment District ID is invalid, must be 1 - 10."
    },
    {
      ERR_PAYMENT_HAM_INVALID,
      "Payment Amount
invalid data type must be numeric."
    },
    {
      ERR_PAYMENT_HAM_RANGE,
      "Payment Amount out of range, 0 - 9999.99."
    },
    {
      ERR_PAYMENT_LAST_NAME_TO_LONG,
      "Payment Customer last name longer than 16 characters."
    },
    {
      ERR_PAYMENT_MISSING_CDI_KEY,
      "Payment missing Customer district key \"CDI*\"."
    },
    {
      ERR_PAYMENT_MISSING_CID_CLT,
      "Payment Either Customer ID or Last Name must be
entered."
    },
    {
      ERR_PAYMENT_MISSING_CID_KEY,
      "Payment missing Customer Key \"CID*\"."
    },
    {
      ERR_PAYMENT_MISSING_CLT_KEY,
      "Payment missing Customer Last Name key \"CLT*\"."
    },
    {
      ERR_PAYMENT_MISSING_CWI_KEY,
      "Payment missing Customer Warehouse key \"CWI*\"."
    },
    {
      ERR_PAYMENT_MISSING_DID_KEY,
      "Payment missing District Key \"DID*\"."
    },
  },

```

```

        {
            ERR_PAYMENT_MISSING_HAM_KEY,
            "Payment missing Amount key \"HAM*\"."
        },
        {
            ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
            "Stock Level; missing Threshold key \"TT*\"."
        },
        {
            ERR_STOCKLEVEL_THRESHOLD_INVALID,
            "Stock Level; Threshold value must be in the range = 1 -
99."
        },
        {
            ERR_STOCKLEVEL_THRESHOLD_RANGE,
            "Stock Level Threshold out of range, range
must be 1 - 99."
        },
        {
            ERR_VERSION_MISMATCH,
            "Invalid version field. RTE and Web Client are probably
out of sync." },
        {
            ERR_W_ID_INVALID,
            "Invalid Warehouse ID."
        },
        {
            0,
            ""
        }
    };

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

```

```

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

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

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

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

```

```

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

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

    *pQueryString = ptr;
    return;

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

/* FUNCTION: GetIntKeyValue
*
* PURPOSE: This function parses a http formatted string for a specific
key value.
*
* ARGUMENTS:      char
                  *pQueryString      http string from client browser
*
*                  char              key
                  *pKey
value to look for
*
*                  WEBERROR
                  NoKeyErr
error value to throw if key not
found
*
*                  WEBERROR
                  NotIntErr
error value to throw if value
not numeric
*
* RETURNS:         integer
*
* ERROR:           if (the pKey value is not found) then
if
(NoKeyErr != NO_ERR)
*
*                 throw CWEBCLNT_ERR(err)
*
*                 else
*
*                 return 0

```

```

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

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

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

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

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

    *pQueryString = ptr;
    return atoi(ptr0);

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

```

```

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

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

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

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

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

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

    LeaveCriticalSection(&TermCriticalSection);
}

/* FUNCTION: TermDeleteAll
*
* PURPOSE: This function frees allocated resources associated with the
terminal structure.
*
* ARGUMENTS:      none
*
* RETURNS:        None
*

```

```

* COMMENTS:      This function is called only when the inet
service unloads the TPCC.DLL
*/

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

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

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

    LeaveCriticalSection(&TermCriticalSection);
}

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

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

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

    EnterCriticalSection(&TermCriticalSection);
    if (Term.iFreeList != 0)
    {
        // position is available
        iNewTerm = Term.iFreeList;
        Term.iFreeList =
Term.pClientData[iNewTerm].iNextFree;
        Term.pClientData[iNewTerm].iNextFree = -
1; // indicates this position is in use
    }
    else

```

```

        {
            // no open slots, so find the slot that hasn't
            been used in the longest time and reuse it
            for(iNewTerm=1, i=1,
iTickCount=0x7FFFFFFF; i<Reg.dwMaxConnections; i++)
            {
                if (iTickCount >

Term.pClientData[i].iTickCount)
                {
                    iTickCount =

Term.pClientData[i].iTickCount;

                    iNewTerm = i;
                }
            }
            // if oldest term is less than one minute old,
            it probably means that more connections
            // are being attempted than were specified
            as "Max Connections" at install. In this case,
            // do not bump existing connection; instead,
            return error to requestor.
            if ((GetTickCount() - iTickCount) < 60000)
            {

                LeaveCriticalSection(&TermCriticalSection);
                throw new CWEBCLNT_ERR(
ERR_MAX_CONNECTIONS_EXCEEDED );
            }

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

            LeaveCriticalSection(&TermCriticalSection);
            return iNewTerm;
        }

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

void TermDelete(int id)
{
    if ( id > 0 && id < Term.iNumEntries )

```

```

        {
            delete Term.pClientData[id].pTxn;

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

            Term.pClientData[id].iNextFree =

Term.iFreeList;

            Term.iFreeList = id;

            LeaveCriticalSection(&TermCriticalSection);
        }

}

/* FUNCTION: MakeErrorForm
*/

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

```

```

, iTType, iErrorNum, MAIN_MENU_FORM,
iTtermId, iSyncId, szErrorText );

/* FUNCTION: MakeMainMenuForm
*/

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

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

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

```



```

        " <INPUT NAME=\"SP10*\"
SIZE=4> <INPUT NAME=\"IID10*\" SIZE=6>
<INPUT NAME=\"Qty10*\" SIZE=1><BR>\"
        " <INPUT NAME=\"SP11*\"
SIZE=4> <INPUT NAME=\"IID11*\" SIZE=6>
<INPUT NAME=\"Qty11*\" SIZE=1><BR>\"
        " <INPUT NAME=\"SP12*\"
SIZE=4> <INPUT NAME=\"IID12*\" SIZE=6>
<INPUT NAME=\"Qty12*\" SIZE=1><BR>\"
        " <INPUT NAME=\"SP13*\"
SIZE=4> <INPUT NAME=\"IID13*\" SIZE=6>
<INPUT NAME=\"Qty13*\" SIZE=1><BR>\"
        " <INPUT NAME=\"SP14*\"
SIZE=4> <INPUT NAME=\"IID14*\" SIZE=6>
<INPUT NAME=\"Qty14*\" SIZE=1><BR>\"
        "Execution Status:
Total:<BR>\"
        "</font></PRE><HR>\"
        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">\"
        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">\"
        "</FORM></HTML>\"
    );
}
else
{
    c += wsprintf(szForm+c, "Warehouse:
%6.6d District: %2.2d Date: ",
pNewOrderData->w_id,
pNewOrderData->d_id);

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

    c += wsprintf(szForm+c, "<BR>Customer:
%4.4d Name: %-16s Credit: %-2s ",

```

```

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

        for(i=0; i<pNewOrderData-
>o_ol_cnt; i++)
        {
            c +=
            sprintf(szForm+c, "%6.6d %6.6d %-24s %2.2d %3.3d %1.1s
$%6.2f $%7.2f <BR>\",
pNewOrderData->OL[i].ol_supply_w_id,
pNewOrderData->OL[i].ol_i_id,
pNewOrderData->OL[i].ol_i_name,
pNewOrderData->OL[i].ol_quantity,
pNewOrderData->OL[i].ol_stock,
pNewOrderData->OL[i].ol_brand_generic,
pNewOrderData->OL[i].ol_i_price,
pNewOrderData->OL[i].ol_amount );
        }
    }
    else
    {
        c += wsprintf(szForm+c,
        "%5.2f <BR>\"

```

```

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

        i = 0;
    }

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

    if ( bValid )
    {
        c += sprintf(szForm+c,
        "Execution Status: Transaction committed. Total: $%8.2f
\",
pNewOrderData->total_amount);
    }
    else
    {
        c += wsprintf(szForm+c,
        "Execution Status: Item number is not valid. Total:");
        strcpy(szForm+c,
        " <BR></font></PRE><HR>\"
        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..NewOrder..\">\"
        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">\"
        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">\"
        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">\"
        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">\"
        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">\"
        "</FORM></HTML>\"
    );
}

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

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

```

```

        c = sprintf(szfForm,
            "<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=\"SYCID\" VALUE=\"%d\">"
            "<PRE><font face=\"Courier\">"
Payment<BR>"
            "Date: "
            ", PAYMENT_FORM, iTermId,
Term.pClientData[iTermId].iSyncId);

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

        if ( bInput )
        {
            c += sprintf(szfForm+c,
                "<BR> <BR>Warehouse:
%6.6d"
                " District:
<INPUT NAME=\"DID*\" SIZE=1><BR> <BR> <BR> <BR> <BR>"
                "Customer: <INPUT
NAME=\"CID*\" SIZE=4>"
                "Cust-Warehouse: <INPUT
NAME=\"CWI*\" SIZE=4> "
                "Cust-District: <INPUT
NAME=\"CDI*\" SIZE=1><BR>"
                "Name:
<INPUT NAME=\"CLT*\" SIZE=16> Since:<BR>"

```

```

            "
            Credit:<BR>"
            Disc:<BR>"
            "
            Phone:<BR> <BR>"
            "Amount Paid: $<INPUT
NAME=\"HAM*\" SIZE=7> New Cust-Balance:<BR>"
            "Credit Limit:<BR> <BR>Cust-
Data: <BR> <BR> <BR> <BR> <BR></font></PRE><HR>"
            "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\"><INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"
            "</BODY></FORM></HTML>"
            ,
            Term.pClientData[iTermId].w_id);
        }
        else
        {
            c += sprintf(szfForm+c,
                "<BR> <BR>Warehouse:
%6.6d District: %2.2d<BR>"
                "%-20s %-
20s<BR>"
                "%-20s %-
20s<BR>"
                "%-20s %-2s %5.5s-%4.4s<BR> <BR>"
                "Customer: %4.4d Cust-
Warehouse: %6.6d Cust-District: %2.2d<BR>"
                "Name: %-16s %-2s %-16s
Since: %2.2d-%2.2d-%4.4d<BR>"
                " %-20s
Credit: %-2s<BR>"
            ,
            Term.pClientData[iTermId].w_id, pPaymentData->d_id
            , pPaymentData->w_street_1,
            pPaymentData->d_street_1
            , pPaymentData->w_street_2,
            pPaymentData->d_street_2
            , pPaymentData->w_city,
            pPaymentData->w_state, pPaymentData->w_zip, pPaymentData-
>w_zip+5
            , pPaymentData->d_city,
            pPaymentData->d_state, pPaymentData->d_zip, pPaymentData-
>d_zip+5
            , pPaymentData->c_id,
            pPaymentData->c_w_id, pPaymentData->c_d_id
            , pPaymentData->c_first,
            pPaymentData->c_middle, pPaymentData->c_last
            , pPaymentData->c_since.day,
            pPaymentData->c_since.month, pPaymentData->c_since.year

```

```

            , pPaymentData->c_street_1,
            );
            c += sprintf(szfForm+c,
                " %-20s
%%Disc: %5.2f<BR>",
                pPaymentData->c_street_2,
                100.0*pPaymentData->c_discount);
            c += sprintf(szfForm+c,
                " %-20s %-2s %5.5s-
%4.4s Phone: %6.6s-%3.3s-%3.3s-%4.4s<BR> <BR>",
                pPaymentData->c_city,
                pPaymentData->c_state, pPaymentData->c_zip, pPaymentData-
>c_zip+5,
                pPaymentData->c_phone,
                pPaymentData->c_phone+6, pPaymentData->c_phone+9,
                pPaymentData->c_phone+12 );
            c += sprintf(szfForm+c,
                "Amount Paid: $%7.2f
New Cust-Balance: $%14.2f<BR>"
                "Credit Limit: $%13.2f<BR>
",
                pPaymentData->h_amount,
                pPaymentData->c_balance
                , pPaymentData->c_credit_lim
            );
            if ( pPaymentData->c_credit[0] == 'B' &&
                pPaymentData->c_credit[1] == 'C' )
            c += sprintf(szfForm+c,
                "Cust-Data: %-50.50s<BR> %-50.50s<BR>
%-50.50s<BR> %-50.50s<BR>",
                pPaymentData->c_data, pPaymentData->c_data+50,
                pPaymentData->c_data+100, pPaymentData->c_data+150 );
            else
                strcpy(szfForm+c, "Cust-Data:
<BR> <BR> <BR> <BR>");
            strcat(szfForm,
                "
<BR></font></PRE><HR>"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"

```

```

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

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

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

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

}

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

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

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

    if ( bInput )

```

```

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

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

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

```

```

            pOrderStatusData-
pOrderStatusData-
>o_entry_d.second,
>o_carrier_id);

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

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

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

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

```

```

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

    c = sprintf(szForm,
        "<HTML><HEAD><TITLE>TPC-C  

        Delivery</TITLE></HEAD><BODY>"
        "<FORM ACTION=\"tpcc.dll\"  

        METHOD=\"GET\">"
        "<INPUT TYPE=\"hidden\"  

        NAME=\"STATUSID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\"  

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

        NAME=\"FORMID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\"  

        NAME=\"TERMINID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\"  

        NAME=\"SYNCID\" VALUE=\"%d\">"
        "<PRE><font face=\"Courier\">"
        Delivery<BR>"
        "Warehouse: %6.6d<BR> <BR>",
        (!bInput && (pDeliveryData-
        >exec_status_code != eOK)) ? ERR_TYPE_DELIVERY_POST : 0,
        DELIVERY_FORM, iTermId,
        Term.pClientData[iTermId].iSyncId, Term.pClientData[iTermId].w_id);

    if ( bInput )
    {
        strcpy( szForm+c,
            "Carrier Number: <INPUT  

            NAME=\"OCD*\" SIZE=1><BR> <BR>"
            "Execution Status: <BR>  

            <BR> <BR> <BR> <BR> <BR> <BR>  

            " <BR> <BR> <BR> <BR>  

            <BR> <BR> <BR> <BR> </font></PRE><HR>"
            "<INPUT TYPE=\"submit\"  

            NAME=\"CMD\" VALUE=\"Process\">"
            "<INPUT TYPE=\"submit\"  

            NAME=\"CMD\" VALUE=\"Menu\">"
            "</BODY></FORM></HTML>" );
    }
    else
    {
        sprintf( szForm+c,
            "Carrier Number: %2.2d<BR>  

            <BR>"
            "Execution Status: %s <BR>  

            <BR> <BR> <BR> <BR> <BR> <BR> <BR>"

```

```

        "<BR> <BR> <BR> <BR>  

        <BR> <BR> <BR> <BR> </font></PRE><HR><INPUT  

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

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

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

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

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

        NAME=\"CMD\" VALUE=\"..Exit..\">"
        "</BODY></FORM></HTML>"
        , pDeliveryData->o_carrier_id,
        (pDeliveryData-
        >exec_status_code == eOK) ? "Delivery has been queued." : "Delivery  

        Post Failed "
        );
    }

    /* FUNCTION: ProcessNewOrderForm
    *
    * PURPOSE: This function gets and validates the input data from the  

    * new order form
    * filling in the required input  

    * variables. it then calls the SQLNewOrder  

    * transaction, constructs the  

    * output form and writes it back to client  

    * browser.
    */

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

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

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

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

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

```

```

    }
    /* FUNCTION: void ProcessPaymentForm
    *
    * PURPOSE: This function gets and validates the input data from the  

    * payment form
    * filling in the required input  

    * variables. It then calls the SQLPayment  

    * transaction, constructs the  

    * output form and writes it back to client  

    * browser.
    *
    * ARGUMENTS: EXTENSION_CONTROL_BLOCK
    * *pECB passed in structure pointer from inetsrv.  

    * int
    *
    * iTermId client browser terminal id
    */

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

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

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

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

    /* FUNCTION: ProcessOrderStatusForm
    *
    * PURPOSE: This function gets and validates the input data from the  

    * Order Status
    * form filling in the required  

    * input variables. It then calls the  

    * SQLOrderStatus transaction,  

    * constructs the output form and writes it  

    * back to client browser.
    *
    * ARGUMENTS: EXTENSION_CONTROL_BLOCK
    * *pECB passed in structure pointer from inetsrv.  

    * int
    *
    * iTermId client browser terminal id
    */

```

```

*
*/

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

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

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

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

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

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

    PDELIVERY_DATA pDelivery;

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

```

```

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

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

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

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

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

```

```

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

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

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

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

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

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

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

    static char szSP[MAX_OL_NEW_ORDER_ITEMS][6] =
{ "SP00*", "SP01*", "SP02*", "SP03*",
"SP04*",
"SP05*", "SP06*", "SP07*", "SP08*",
"SP09*",

```

```

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

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

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

                        ol_i_id = pNewOrderData-
>OL[items].ol_i_id =

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

                                ol_quantity = pNewOrderData-
>OL[items].ol_quantity =

```

```

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

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

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

                pNewOrderData->o_ol_cnt = items;
}

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

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

```

```

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

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

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

                _strup( 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);
    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
//
#define 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, "Unspecified") )
            pReg->eDB_Protocol = ODBC;
        else if ( !strcmp(szTmp, "ODBC") )
            pReg->eDB_Protocol = ODBC;
        else if ( !strcmp(szTmp, "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, "TUXEDO") )
            pReg->eTxnMon = TUXEDO;
        else if ( !strcmp(szTmp, "ENCINA") )
            pReg->eTxnMon = ENCINA;
        else if ( !strcmp(szTmp, "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
*
*      Copyright
*
*      Microsoft, 1999
*      All Rights Reserved
*
*      Version 4.10.000
*      audited by Richard Gimarc, Performance Metrics, 3/17/99
*
*      PURPOSE:   Header file for error exception classes.
*
*      Change history:
*      4.20.000 - updated rev number to match kit

```

```

*/
#pragma once

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

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

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

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

#define ERR_TYPE_LOGIC           -1
                        //logic error in program; internal error
#define ERR_SUCCESS              0
                        //success (a non-error error)
#define ERR_BAD_ITEM_ID         1
                        //expected abort record in txnRecord
#define ERR_TYPE_DELIVERY_POST  2
                        //expected
                        delivery post failed
#define ERR_TYPE_WEBDLL          3
                        //tpcc web generated error
#define ERR_TYPE_SQL             4
                        //sql server generated error
#define ERR_TYPE_DBLIB           5
                        //dblib generated error
#define ERR_TYPE_ODBC            6
                        //odbc generated error
#define ERR_TYPE_SOCKET          7
                        //error on communication socket client rte only

```

```

#define ERR_TYPE_DEADLOCK        8
#define ERR_TYPE_TPCC_ODBC_ONLY 9
                        //error from COM call
#define ERR_TYPE_TUXEDO          10
                        //tuxedo error
#define ERR_TYPE_OS              11
                        //operating system error
#define ERR_TYPE_MEMORY          12
                        //memory allocation error
#define ERR_TYPE_TPCC_ODBC       13
                        //error from tpcc odbc txn module
#define ERR_TYPE_TPCC_DBLIB      14
                        //error from tpcc dblib txn module
#define ERR_TYPE_DELISRV         15
                        //delivery server error
#define ERR_TYPE_TXNLOG          16
                        //txn log error
#define ERR_TYPE_BCCONN          17
                        //Benchcraft connection class
#define ERR_TYPE_TPCC_CONN       18
                        //Benchcraft connection class
#define ERR_TYPE_ENCINA          19
                        //Encina error
#define ERR_TYPE_COMPONENT        20
                        //error from COM component

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

    CBaseErr(void)
    {
        m_idMsg          = 0;
        m_szMsg           = new
char[m_szMsg_size];

```

```

        m_szApp          = new
char[m_szApp_size];
        m_szLoc          = NULL;

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

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

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

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

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

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

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

```

```

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

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

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

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

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

char *GetApp(void) { return m_szApp; }
char *GetMsg(void) { return m_szMsg; }
char *GetLoc(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      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 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.
#ifdef __SQLTYPES
    typedef struct
    {
        /* SQLSMALLINT */      short
        /* SQLSMALLINT */      year;
        /* SQLSMALLINT */      unsigned short
        /* SQLSMALLINT */      month;
        /* SQLSMALLINT */      unsigned short
        /* SQLSMALLINT */      day;
        /* SQLSMALLINT */      unsigned short
        /* SQLSMALLINT */      hour;
        /* SQLSMALLINT */      unsigned short
        /* SQLSMALLINT */      minute;
        /* SQLSMALLINT */      unsigned short
        /* SQLSMALLINT */      second;
        /* SQLSMALLINT */      unsigned long
        /* SQLINTEGER */        fraction;
    } TIMESTAMP_STRUCT;
#else
    typedef struct
    {
        /* SQLSMALLINT */      short
        /* SQLSMALLINT */      year;
        /* SQLSMALLINT */      unsigned short
        /* SQLSMALLINT */      month;
        /* SQLSMALLINT */      unsigned short
        /* SQLSMALLINT */      day;
        /* SQLSMALLINT */      unsigned short
        /* SQLSMALLINT */      hour;
        /* SQLSMALLINT */      unsigned short
        /* SQLSMALLINT */      minute;
        /* SQLSMALLINT */      unsigned short
        /* SQLSMALLINT */      second;
        /* SQLSMALLINT */      unsigned long
        /* SQLINTEGER */        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

```

```

} PAYMENT_DATA, *PPAYMENT_DATA;          c_data[200+1];

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;

    // output params
    EXEC_STATUS
    exec_status_code;
    SYSTEMTIME
    long          queue_time;
    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;
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:          TPCC_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 <stdio.h>
#include <assert.h>

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

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

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

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

#define DEFCLPACKSIZE
4096

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

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

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

BOOL WINAPI DllMain(HMODULE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved)
{
    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:

            DisableThreadLibraryCalls(hModule);
            dbinit(); //

            initialize dblib

            break;

        case DLL_PROCESS_DETACH:
            dbexit(); //

            close all dblib structures/connections

            break;

        default:
            /* nothing */;
    }
}

```

```

        return TRUE;
    }

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

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

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

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

```

```

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

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

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

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

    return 0;
}

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

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

```

```

    return;
}

/* FUNCTION: CTPCC_DBLIB::ErrorText
*
*/

char* CTPCC_DBLIB::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 ) // 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);

DBSETUSER(login, szUser);
DBSETLPWD(login, szPassword);
DBSETHOST(login, szHost);
DBSETPACKET(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 (dbsqlxexec(m_dbproc) == FAIL)
    ThrowError(CDBLIBERR::eDbSqlExec);

DiscardNextResults(2);

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

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

```

```

if (dbresults(m_dbproc, CDBLIBERR::eDbResults);
    ThrowError(CDBLIBERR::eDbResults);

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

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

DiscardNextRows(0);
DiscardNextResults(0);
}

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

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

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

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

```

```

}

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

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

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

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

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

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

    throw pDbLibErr;
}

```

```

}

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

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

                ThrowError(CDBLIBERR::eDbNextRow);
            else
                break;

        }
        iRowsRead++;
    }

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

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

    while (TRUE)
    {
        rc = dbresults(m_dbproc);
        if (rc == NO_MORE_RESULTS)
            break;
    }

```

```

        if (rc == FAIL)
            if (iExpectedCount >= 0)

                ThrowError(CDBLIBERR::eDbResults);
            else
                break;

        }

        DiscardNextRows(-1);
        iResultsRead++;
    }

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

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

    ResetError();

    while (TRUE)
    {
        try
        {
            dbrpcinit(m_dbproc,

                "tpcc_stocklevel", 0);

            dbrpcparam(m_dbproc, NULL,

                0, 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 &&
                            strstr(e-
                                >m_msgtext, sErrTimeoutExpired) != NULL)) &&
                                (++iTryCount <=
                                    iMaxRetries))
                    {
                        // hit deadlock;
                        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;
            DBDATETIME 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
(dbnextrrow(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 (dbnextrrow(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));

```

```

5))          if (pData=dbdata(m_dbproc,

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

UtilStrCpy(m_txn.NewOrder.c_credit, pData,
dbdatlen(m_dbproc, 6));
7))          if (pData=dbdata(m_dbproc,
{
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 =

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 &&

>m_msgtext, 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::Payment()
{
DBDATETIME datetime;
DBDATERECD daterec;

int
const BYTE *pData;

iTryCount = 0;

ResetError();

while (TRUE)
{
try
{
dbrpcinit(m_dbproc,

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.h_date);
dbrpcparam(m_dbproc, NULL,
0, SQLINT1, -1, -1, (BYTE *) &m_txn.Payment.c_d_id);
dbrpcparam(m_dbproc, NULL,
0, SQLINT4, -1, -1, (BYTE *) &m_txn.Payment.c_id);

// if customer id is zero, then
payment is by name
if (m_txn.Payment.c_id == 0)

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

if (dbrpcexec(m_dbproc) ==
FAIL)

ThrowError(CDBLIBERR::eDbRpcExec);

if (dbresults(m_dbproc) !=
SUCCEED)

ThrowError(CDBLIBERR::eDbResults);

if (dbnextrow(m_dbproc) !=
REG_ROW)

ThrowError(CDBLIBERR::eDbNextRow);

if (dbnumcols(m_dbproc) !=
27)

ThrowError(CDBLIBERR::eWrongNumCols);

if (pData=dbdata(m_dbproc,
1))

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

UtilStrCpy(m_txn.Payment.c_last, pData,
dbdatlen(m_dbproc, 2));
if (pData=dbdata(m_dbproc,
3))
{
datetime =
*((DBDATETIME *) pData);

dbdatecrack(m_dbproc, &daterec, &datetime);

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

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

```

```

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

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

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

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

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

    if (pData=dbdata(m_dbproc,
5))

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

    if (pData=dbdata(m_dbproc,
6))

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

    if (pData=dbdata(m_dbproc,
7))

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

    if (pData=dbdata(m_dbproc,
8))

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

    if (pData=dbdata(m_dbproc,
9))

        UtilStrCpy(m_txn.Payment.d_street_1, pData,
dbdatlen(m_dbproc, 9));

    if (pData=dbdata(m_dbproc,
10))

        UtilStrCpy(m_txn.Payment.d_street_2, pData,
dbdatlen(m_dbproc, 10));

    if (pData=dbdata(m_dbproc,
11))

        UtilStrCpy(m_txn.Payment.d_city, pData,
dbdatlen(m_dbproc, 11));

    if (pData=dbdata(m_dbproc,
12))

```

```

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

        UtilStrCpy(m_txn.Payment.d_zip, pData,
dbdatlen(m_dbproc, 13));

    if (pData=dbdata(m_dbproc,
14))

        UtilStrCpy(m_txn.Payment.c_first, pData,
dbdatlen(m_dbproc, 14));

    if (pData=dbdata(m_dbproc,
15))

        UtilStrCpy(m_txn.Payment.c_middle, pData,
dbdatlen(m_dbproc, 15));

    if (pData=dbdata(m_dbproc,
16))

        UtilStrCpy(m_txn.Payment.c_street_1, pData,
dbdatlen(m_dbproc, 16));

    if (pData=dbdata(m_dbproc,
17))

        UtilStrCpy(m_txn.Payment.c_street_2, pData,
dbdatlen(m_dbproc, 17));

    if (pData=dbdata(m_dbproc,
18))

        UtilStrCpy(m_txn.Payment.c_city, pData,
dbdatlen(m_dbproc, 18));

    if (pData=dbdata(m_dbproc,
19))

        UtilStrCpy(m_txn.Payment.c_state, pData,
dbdatlen(m_dbproc, 19));

    if (pData=dbdata(m_dbproc,
20))

        UtilStrCpy(m_txn.Payment.c_zip, pData,
dbdatlen(m_dbproc, 20));

    if (pData=dbdata(m_dbproc,
21))

        UtilStrCpy(m_txn.Payment.c_phone, pData,
dbdatlen(m_dbproc, 21));

    if (pData=dbdata(m_dbproc,
22))

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

```

```

        dbdatecrack(m_dbproc, &daterec, &datetime);
        m_txn.Payment.c_since.year = daterec.year;

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

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

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

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

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

        UtilStrCpy(m_txn.Payment.c_credit, pData,
dbdatlen(m_dbproc, 23));

    if (pData=dbdata(m_dbproc,
24))

        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,24), SQLFLT8, (BYTE
*)&m_txn.Payment.c_credit_lim, 8);

    if (pData=dbdata(m_dbproc,
25))

        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,25), SQLFLT8, (BYTE
*)&m_txn.Payment.c_discount, 8);

    if (pData=dbdata(m_dbproc,
26))

        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,26), SQLFLT8, (BYTE
*)&m_txn.Payment.c_balance, 8);

    if (pData=dbdata(m_dbproc,
27))

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

        DiscardNextRows(0);
        DiscardNextResults(0);

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

        m_txn.Payment.exec_status_code = eOK;

```



```

        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 &&
>m_msgtext, sErrTimeoutExpired) != NULL)) &&
                (++iTryCount <=
iMaxRetries))
            {
                // hit deadlock;
                backoff for increasingly longer period

                delete e;
                Sleep(10 *
iTryCount);
            }
            else
                throw;

```

```

        } // while (TRUE)

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

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

        ResetError();

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

                dbrpcparam(m_dbproc, NULL,
0, 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 &&
            >m_msgtext, sErrTimeoutExpired) != NULL)) &&
        iMaxRetries))
    {
        // hit deadlock;
        backoff for increasingly longer period
        iTryCount);
        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 <sqlext.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 WINAPI 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)

                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 name for login
    LPCSTR szPassword,        // password for
login
    LPCSTR szHost,            //
not used
    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
    // 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 ==
iErrOleDbProvider &&
sErrTimeoutExpired) != NULL))
            strncat(szMsg,

```

```

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 error text
        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,
SQL_CLOSE);

            m_txn.StockLevel.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::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_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
    1      2

    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_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)
            {
                m_txn.NewOrder.o_all_local =
0; // at least one remote warehouse
                break;
            }
        }
        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 value...
                    m_BindOffset = i
* sizeof(m_txn.NewOrder.OL[0]);

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

                    // move to the
next resultset
                    if (
SQLMoreResults(m_hstmt) == SQL_ERROR )
                        ThrowError(CODBCERR::eMoreResults);

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

```

```

        // 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,
&m_txn.Payment.c_last, 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,
&m_txn.Payment.w_street_1, sizeof(m_txn.Payment.w_street_1),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_street_2, sizeof(m_txn.Payment.w_street_2),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_city, sizeof(m_txn.Payment.w_city),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_state, sizeof(m_txn.Payment.w_state),
NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindCol);
}

```

```

        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,
&m_txn.Payment.c_last, 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,
&m_txn.Payment.w_street_1, sizeof(m_txn.Payment.w_street_1),
NULL) != SQL_SUCCESS
            || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_street_2, sizeof(m_txn.Payment.w_street_2),
NULL) != SQL_SUCCESS
            || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_city, sizeof(m_txn.Payment.w_city),
NULL) != SQL_SUCCESS
            || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_state, sizeof(m_txn.Payment.w_state),
NULL) != SQL_SUCCESS
        )
            ThrowError(CODBCERR::eBindCol);
    }
}

```

```

        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_zip,
        sizeof(m_txn.Payment.w_zip),
        NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_street_1,
        sizeof(m_txn.Payment.d_street_1), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_street_2,
        sizeof(m_txn.Payment.d_street_2), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_city,
        sizeof(m_txn.Payment.d_city),
        NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_state,
        sizeof(m_txn.Payment.d_state), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_zip,
        sizeof(m_txn.Payment.d_zip),
        NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_first,
        sizeof(m_txn.Payment.c_first),
        NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_middle,
        sizeof(m_txn.Payment.c_middle), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_street_1,
        sizeof(m_txn.Payment.c_street_1), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_street_2,
        sizeof(m_txn.Payment.c_street_2), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_city,
        sizeof(m_txn.Payment.c_city),
        NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_state,
        sizeof(m_txn.Payment.c_state),
        NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_zip,
        sizeof(m_txn.Payment.c_zip),
        NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_phone,
        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,
&m_txn.Payment.c_credit, 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,
&m_txn.Payment.c_data,
        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 =
SQLExecDirectW(m_hstmt, (SQLWCHAR*)"{"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_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 (!(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::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*)"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 (!(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);
}

```

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.
#ifdef 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
    };

    CODBCERR(void)
    {
        m_eAction = eNone;
        m_NativeError = 0;
        m_bDeadLock = FALSE;
        m_odbcerrstr = NULL;
    };

    ~CODBCERR()
    {
        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;
    // ODBC environment handle
    SQLHDBC        m_hdbc;
    // 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( CODBCERR::ACTION
eAction );

void InitNewOrderParams();
void InitPaymentParams();
void InitDeliveryParams();
void InitStockLevelParams();
void InitOrderStatusParams();

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

    // use this interface to impersonate a non-
COM error type
    CCOMERR( int iErrorType, int iError )
    {
        m_iErrorType = iErrorType;
        m_iError = iError;
        m_hr = S_OK;
    }

    int m_hr;
    int m_iErrorType;
    int m_iError;

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

    int ErrorNum() {return m_hr;}

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

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

    // COM Interface pointers
    ITPCC*
m_pNewOrder;
```

```
    ITPCC*
m_pPayment;ITPCC*
m_pStockLevel;
    ITPCC*
m_pOrderStatus;

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

public:
    CTPCC_COM(BOOL bSinglePool);
    ~CTPCC_COM(void);

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

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

```

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

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

typedef CTPCC_COM* (TYPE_CTPCC_COM)(BOOL);

```

tm_com_dll\src\tpcc_com.cpp

```

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

// needed for CoInitializeEx
#define _WIN32_WINNT 0x0400

#include <windows.h>

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

#include "..\..\common\src\trans.h" //tpckit
transaction header contains 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*/
#ifdef __REQUIRED_RPCNDR_H_VERSION__

```

```

#ifdef __REQUIRED_RPCNDR_H_VERSION__ 440

```

```

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

```

```

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

```

```

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

```

```

#ifdef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

```

```

/* Forward Declarations */

```

```

#ifdef __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;

```

```

#ifdef __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 ( __stdcall __RPC_FAR *NewOrder )(
ITPCC __RPC_FAR * This,
/* [in] */ VARIANT txn_in,
/* [out] */ VARIANT __RPC_FAR *txn_out);
```

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

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

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

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

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

```
END_INTERFACE
} ITPCCVtbl;
```

```
interface ITPCC
```

```
{
CONST_VTBL struct ITPCCVtbl __RPC_FAR *lpVtbl;
};
```

```
#ifdef COBJMACROS
```

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

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

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

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

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

```
#define ITPCC_Delivery(This,txn_in,txn_out) \
```

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

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

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

```
#endif /* COBJMACROS */
```

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

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

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

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

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

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

```
void __RPC_STUB ITPCC_Delivery_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *_pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
```

```

DWORD *_pdwStubPhase);

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

```

```

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

```

```

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

```

```

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

```

```

HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC __RPC_FAR * This);

```

```

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

```

```

#endif /* __ITPCC_INTERFACE_DEFINED__ */

```

```

/* Additional Prototypes for ALL interfaces */

```

```

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

```

```

unsigned char __RPC_FAR * __RPC_USER
VARIANT_UserUnmarshal(unsigned long __RPC_FAR *, unsigned char
__RPC_FAR *, VARIANT __RPC_FAR *);
unsigned long __RPC_FAR * __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
```

```
typedef IID CLSID;
```

```
#endif // CLSID_DEFINED
```

```
#define
```

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

```
const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}
```

```
#endif !_MIDL_USE_GUIDDEF_
```

```
MIDL_DEFINE_GUID(IID,
```

```
IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,
```

```
0xE0,0x8B);
```

```
#undef MIDL_DEFINE_GUID
```

```
#ifdef __cplusplus
```

```
}
```

```
#endif
```

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

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

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

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

```
/* File created by MIDL compiler version 5.03.0280 */
```

```
/* 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, allocation ref bounds_check enum stub_data
```

```
VC __declspec() decoration level:
```

```
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
```

```
DECLSPEC_UUID(), MIDL_INTERFACE()
```

```
*/
```

```
//@@MIDL_FILE_HEADING( )
```

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

```
#ifdef __cplusplus
```

```
extern "C"{
```

```
#endif
```

```
#include <rpc.h>
```

```
#include <rpcndr.h>
```

```
#ifdef _MIDL_USE_GUIDDEF_
```

```
#ifndef INITGUID
```

```
#define INITGUID
```

```
#include <guiddef.h>
```

```
#undef INITGUID
```

```
#else
```

```
#include <guiddef.h>
```

```
#endif
```

```
#define
```

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

```
DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)
```

```
#else // !_MIDL_USE_GUIDDEF_
```

```
#ifndef __IID_DEFINED__
```

```
#define __IID_DEFINED__
```

```
typedef struct _IID
```

```
{
```

```
unsigned long x;
```

```
unsigned short s1;
```

```
unsigned short s2;
```

```
unsigned char c[8];
```

```
} IID;
```

```
#endif // __IID_DEFINED__
```

```
#ifndef CLSID_DEFINED
```

```
#define CLSID_DEFINED
```

```
typedef IID CLSID;
```

```
#endif // CLSID_DEFINED
```

```

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

#endif !_MIDL_USE_GUIDDEF_

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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

tpcc_com_ps\src\tpcc_com_ps_p.c

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

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

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

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

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

```

```

#endif

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

#include "tpcc_com_ps.h"

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

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

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

extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00}} */

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

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */

extern const MIDL_STUB_DESC Object_StubDesc;

```

```

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

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

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

INTERFACE_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,
    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 defined(_MIPS_)
#else
MIPS Stack size/offset = 8 */
#endif
#else
PPC Stack size/offset = 8 */
#endif
#else
Alpha Stack size/offset = 8 */
#endif
/* 20 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */
/* Parameter txn_out */
/* 22 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free,
out, simple ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
if !defined(_MIPS_)
/* 24 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20
*/
#else
NdrFcShort( 0x18 ), /*
MIPS Stack size/offset = 24 */
#endif
#else
PPC Stack size/offset = 24 */
#endif
#else
Alpha Stack size/offset = 24 */
#endif
/* 26 */ NdrFcShort( 0x3da ), /* Type Offset=986 */
/* Return value */
/* 28 */ NdrFcShort( 0x70 ), /* Flags: out, return, base
type, */
#ifndef _ALPHA_
#ifndef _PPC_
if !defined(_MIPS_)
/* 30 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24
*/
#else
NdrFcShort( 0x1c ), /*
MIPS Stack size/offset = 28 */
#endif
#else

```


#else			0x0,	/* 0	/* 122 */ NdrFcShort(0x3c8), /* Type Offset=968 */
Alpha Stack size/offset = 8 */	NdrFcShort(0x8), /*	*/	/* Procedure StockLevel */		/* 124 */ NdrFcShort(0x4113), /* Flags: must size, must free,
#endif					out, simple ref, srv alloc size=16 */
/* 88 */ NdrFcShort(0x3c8), /* Type Offset=968 */		/* 102 */ 0x33,	/* FC_AUTO_HANDLE */	/*	#ifndef _ALPHA_
/* Parameter txn_out */			0x6c,	/*	#ifndef _PPC_
/* 90 */ NdrFcShort(0x4113), /* Flags: must size, must free,		Old Flags: object, Oi2 */			#if !defined(_MIPS_)
out, simple ref, srv alloc size=16 */		/* 104 */ NdrFcLong(0x0), /* 0 */			/* 126 */ NdrFcShort(0x14), /* x86 Stack size/offset = 20
#ifndef _ALPHA_		/* 108 */ NdrFcShort(0x6), /* 6 */			*/
#ifndef _PPC_		#ifndef _ALPHA_			#else
#if !defined(_MIPS_)		#ifndef _PPC_			MIPS Stack size/offset = 24 */
/* 92 */ NdrFcShort(0x14), /* x86 Stack size/offset = 20		#if !defined(_MIPS_)			#endif
/		/ 110 */ NdrFcShort(0x1c), /* x86 Stack size/offset = 28			#else
#else		*/			
	NdrFcShort(0x18), /*	#else	NdrFcShort(0x20), /*		
MIPS Stack size/offset = 24 */		MIPS Stack size/offset = 32 */			PPC Stack size/offset = 24 */
#endif		#endif			#endif
#else		#else	NdrFcShort(0x20), /*		
	NdrFcShort(0x18), /*	PPC Stack size/offset = 32 */			Alpha Stack size/offset = 24 */
		#endif			#endif
		#else	NdrFcShort(0x28), /*		/* 128 */ NdrFcShort(0x3da), /* Type Offset=986 */
	NdrFcShort(0x18), /*	Alpha Stack size/offset = 40 */			/* Return value */
		#endif			
Alpha Stack size/offset = 24 */		/* 112 */ NdrFcShort(0x0), /* 0 */			/* 130 */ NdrFcShort(0x70), /* Flags: out, return, base
#endif		/* 114 */ NdrFcShort(0x8), /* 8 */			type, */
/* 94 */ NdrFcShort(0x3da), /* Type Offset=986 */		/* 116 */ 0x7,	/* Oi2 Flags: srv must size, clt		#ifndef _ALPHA_
/* Return value */		must size, has return, */	0x3,	/* 3	#ifndef _PPC_
					#if !defined(_MIPS_)
/* 96 */ NdrFcShort(0x70), /* Flags: out, return, base		*/			/* 132 */ NdrFcShort(0x18), /* x86 Stack size/offset = 24
type, */		/* Parameter txn_in */			*/
#ifndef _ALPHA_					#else
#ifndef _PPC_					MIPS Stack size/offset = 28 */
#if !defined(_MIPS_)		/* 118 */ NdrFcShort(0x8b), /* Flags: must size, must free,			#endif
/* 98 */ NdrFcShort(0x18), /* x86 Stack size/offset = 24		in, by val, */			#else
*/		#ifndef _ALPHA_			
#else		#ifndef _PPC_			PPC Stack size/offset = 28 */
	NdrFcShort(0x1c), /*	#if !defined(_MIPS_)			#endif
MIPS Stack size/offset = 28 */		/* 120 */ NdrFcShort(0x4), /* x86 Stack size/offset = 4 */			#else
#endif		#else	NdrFcShort(0x8), /*		
#else					Alpha Stack size/offset = 32 */
	NdrFcShort(0x1c), /*	MIPS Stack size/offset = 8 */			#endif
PPC Stack size/offset = 28 */		#endif	NdrFcShort(0x8), /*		/* 134 */ 0x8,
#endif		#else			/* FC_LONG */
#else					0x0,
	NdrFcShort(0x20), /*	PPC Stack size/offset = 8 */			/* 0
Alpha Stack size/offset = 32 */		#endif			
#endif		#else			/* Procedure OrderStatus */
/* 100 */ 0x8,	/* FC_LONG */	Alpha Stack size/offset = 8 */	NdrFcShort(0x8), /*		/* 136 */ 0x33,
		#endif			/* FC_AUTO_HANDLE */

<pre> Old Flags: object, Oi2 */ /* 138 */ NdrFcLong(0x0), /* 142 */ NdrFcShort(0x7), #ifdef _ALPHA_ #ifdef _PPC_ #if !defined(_MIPS_) /* 144 */ NdrFcShort(0x1c), */ #else MIPS Stack size/offset = 32 */ #endif #else PPC Stack size/offset = 32 */ #endif #else Alpha Stack size/offset = 40 */ #endif /* 146 */ NdrFcShort(0x0), /* 148 */ NdrFcShort(0x8), /* 150 */ 0x7, must size, has return, */ */ 0x3, /* 3 /* Parameter txn_in */ /* 152 */ NdrFcShort(0x8b), in, by val, */ #ifdef _ALPHA_ #ifdef _PPC_ #if !defined(_MIPS_) /* 154 */ NdrFcShort(0x4), #else MIPS Stack size/offset = 8 */ #endif #else PPC Stack size/offset = 8 */ #endif #else Alpha Stack size/offset = 8 */ #endif /* 156 */ NdrFcShort(0x3c8), /* Type Offset=968 */ /* Parameter txn_out */ </pre>	<pre> 0x6c, /* /* 138 */ NdrFcLong(0x0), /* 142 */ NdrFcShort(0x7), /* 144 */ NdrFcShort(0x1c), /* x86 Stack size/offset = 28 */ NdrFcShort(0x20), /* MIPS Stack size/offset = 32 */ NdrFcShort(0x20), /* PPC Stack size/offset = 32 */ NdrFcShort(0x28), /* Alpha Stack size/offset = 40 */ /* 146 */ NdrFcShort(0x0), /* 148 */ NdrFcShort(0x8), /* 150 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has return, */ 0x3, /* 3 /* Parameter txn_in */ /* 152 */ NdrFcShort(0x8b), /* Flags: must size, must free, in, by val, */ #ifdef _ALPHA_ #ifdef _PPC_ #if !defined(_MIPS_) /* 154 */ NdrFcShort(0x4), /* x86 Stack size/offset = 4 */ #else MIPS Stack size/offset = 8 */ NdrFcShort(0x8), /* PPC Stack size/offset = 8 */ NdrFcShort(0x8), /* Alpha Stack size/offset = 8 */ NdrFcShort(0x8), /* /* Type Offset=968 */ /* Parameter txn_out */ </pre>	<pre> /* 158 */ NdrFcShort(0x4113), /* Flags: must size, must free, must size, must free, /* 160 */ NdrFcShort(0x14), /* x86 Stack size/offset = 20 */ MIPS Stack size/offset = 24 */ PPC Stack size/offset = 24 */ Alpha Stack size/offset = 24 */ /* 162 */ NdrFcShort(0x3da), /* Type Offset=986 */ /* Return value */ /* 164 */ NdrFcShort(0x70), /* Flags: out, return, base type, */ #ifdef _ALPHA_ #ifdef _PPC_ #if !defined(_MIPS_) /* 166 */ NdrFcShort(0x18), /* x86 Stack size/offset = 24 */ #else MIPS Stack size/offset = 28 */ PPC Stack size/offset = 28 */ Alpha Stack size/offset = 32 */ /* 168 */ 0x8, /* FC_LONG */ 0x0, /* 0 /* Procedure CallSetComplete */ /* 170 */ 0x33, /* FC_AUTO_HANDLE */ 0x6c, /* Old Flags: object, Oi2 */ /* 172 */ NdrFcLong(0x0), /* 176 */ NdrFcShort(0x8), /* 8 */ #ifdef _ALPHA_ </pre>	<pre> /* 178 */ NdrFcShort(0x8), /* x86, MIPS, PPC Stack size/offset = 8 */ NdrFcShort(0x10), /* Alpha Stack size/offset = 16 */ /* 180 */ NdrFcShort(0x0), /* 182 */ NdrFcShort(0x8), /* 184 */ 0x4, /* Oi2 Flags: has return, */ 0x1, /* 1 /* Return value */ /* 186 */ NdrFcShort(0x70), /* Flags: out, return, base type, */ #ifdef _ALPHA_ /* 188 */ NdrFcShort(0x4), /* x86, MIPS, PPC Stack size/offset = 4 */ #else Alpha Stack size/offset = 8 */ /* 190 */ 0x8, /* FC_LONG */ 0x0, /* 0 } }; static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString = { 0, { /* /* 2 */ 0x12, 0x0, /* FC_UP */ /* Offset= 944 (948) */ 0x2b, /* FC_NON_ENCAPSULATED_UNION */ 0x9, /* FC_ULONG */ /* 8 */ 0x7, /* Corr desc: FC_USHORT */ 0x0, /* /* 10 */ NdrFcShort(0xffff), /* -8 */ /* 12 */ NdrFcShort(0x2), /* Offset= 2 (14) */ /* 14 */ NdrFcShort(0x10), /* 16 */ /* 16 */ NdrFcShort(0x2b), /* 43 */ /* 18 */ NdrFcLong(0x3), /* 3 */ </pre>
--	--	---	---

```

/* 22 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG
*/
/* 24 */ NdrFcLong( 0x11 ), /* 17 */
/* 28 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE
*/
/* 30 */ NdrFcLong( 0x2 ), /* 2 */
/* 34 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT
*/
/* 36 */ NdrFcLong( 0x4 ), /* 4 */
/* 40 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT
*/
/* 42 */ NdrFcLong( 0x5 ), /* 5 */
/* 46 */ NdrFcShort( 0x800c ), /* Simple arm type:
FC_DOUBLE */
/* 48 */ NdrFcLong( 0xb ), /* 11 */
/* 52 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT
*/
/* 54 */ NdrFcLong( 0xa ), /* 10 */
/* 58 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG
*/
/* 60 */ NdrFcLong( 0x6 ), /* 6 */
/* 64 */ NdrFcShort( 0xd6 ), /* Offset= 214 (278) */
/* 66 */ NdrFcLong( 0x7 ), /* 7 */
/* 70 */ NdrFcShort( 0x800c ), /* Simple arm type:
FC_DOUBLE */
/* 72 */ NdrFcLong( 0x8 ), /* 8 */
/* 76 */ NdrFcShort( 0xd0 ), /* Offset= 208 (284) */
/* 78 */ NdrFcLong( 0xd ), /* 13 */
/* 82 */ NdrFcShort( 0xe2 ), /* Offset= 226 (308) */
/* 84 */ NdrFcLong( 0x9 ), /* 9 */
/* 88 */ NdrFcShort( 0xee ), /* Offset= 238 (326) */
/* 90 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 94 */ NdrFcShort( 0xfa ), /* Offset= 250 (344) */
/* 96 */ NdrFcLong( 0x24 ), /* 36 */
/* 100 */ NdrFcShort( 0x308 ), /* Offset= 776 (876) */
/* 102 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 106 */ NdrFcShort( 0x302 ), /* Offset= 770 (876) */
/* 108 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 112 */ NdrFcShort( 0x300 ), /* Offset= 768 (880) */
/* 114 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 118 */ NdrFcShort( 0x2fe ), /* Offset= 766 (884) */
/* 120 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 124 */ NdrFcShort( 0x2fc ), /* Offset= 764 (888) */
/* 126 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 130 */ NdrFcShort( 0x2fa ), /* Offset= 762 (892) */
/* 132 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 136 */ NdrFcShort( 0x2f8 ), /* Offset= 760 (896) */
/* 138 */ NdrFcLong( 0x400b ), /* 16395 */
/* 142 */ NdrFcShort( 0x2e6 ), /* Offset= 742 (884) */
/* 144 */ NdrFcLong( 0x400a ), /* 16394 */
/* 148 */ NdrFcShort( 0x2e4 ), /* Offset= 740 (888) */
/* 150 */ NdrFcLong( 0x4006 ), /* 16390 */

```

```

/* 154 */ NdrFcShort( 0x2e2 ), /* Offset= 746 (900) */
/* 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 */ 0x5b, /*
*/
FC_END */
/* 286 */ NdrFcShort( 0xc ), /* FC_UP */
/* 288 */ /* Offset= 12 (298) */
0x1b, /*
*/
FC_CARRAY */
0x1, /* 1
*/
*/
/* 290 */ NdrFcShort( 0x2 ), /* 2 */
/* 292 */ 0x9, /* Corr desc: FC_ULONG */
0x0, /*
*/
/* 294 */ NdrFcShort( 0xfffc ), /* -4 */
/* 296 */ 0x6, /* FC_SHORT */
0x5b, /*
*/
FC_END */
/* 298 */
0x17, /*
*/
FC_CSTRUCT */
0x3, /* 3
*/
*/
/* 300 */ NdrFcShort( 0x8 ), /* 8 */
/* 302 */ NdrFcShort( 0xffffffff2 ), /* Offset= -14 (288) */
/* 304 */ 0x8, /* FC_LONG */
0x8, /*
*/
FC_LONG */
/* 306 */ 0x5c, /* FC_PAD */
0x5b, /*
*/
FC_END */
/* 308 */
0x2f, /*
*/
FC_IP */
0x5a, /*
*/
FC_CONSTANT_IID */
/* 310 */ NdrFcLong( 0x0 ), /* 0 */
/* 314 */ NdrFcShort( 0x0 ), /* 0 */
/* 316 */ NdrFcShort( 0x0 ), /* 0 */
/* 318 */ 0xc0, /* 192 */
0x0, /* 0
*/
*/
/* 320 */ 0x0, /* 0 */
0x0, /* 0
*/
*/
/* 322 */ 0x0, /* 0 */
0x0, /* 0
*/
*/
/* 324 */ 0x0, /* 0 */
0x46, /*
*/
70 */
/* 326 */
0x2f, /*
*/
FC_IP */

```

FC_CONSTANT_IID */	0x5a,	/*	/* 420 */	0x1b,	/*	/* 460 */	NdrFcShort(0x4f),	/* 4 */
/* 328 */ NdrFcLong(0x20400),	/* 132096 */		FC_CARRAY */	0x3,	/* 3	/* 464 */	NdrFcShort(0xfffffd4),	/* Offset= -44 (420) */
/* 332 */ NdrFcShort(0x0),	/* 0 */		/*			/* 466 */		
/* 334 */ NdrFcShort(0x0),	/* 0 */		/* 422 */ NdrFcShort(0x4),	/* 4 */		FC_END */		0x5b,
/* 336 */ 0xc0,	/* 192 */	/* 0	/* 424 */ 0x19,	/* Corr desc: field pointer,				
/	0x0,	/ 0	FC_ULONG */					0x8,
/* 338 */ 0x0,	/* 0 */	/* 0	/*	0x0,	/*	FC_LONG */		/*
/	0x0,	/ 0	/* 426 */ NdrFcShort(0x0),	/* 0 */		/* 468 */ 0x8,	/* FC_LONG */	/*
/* 340 */ 0x0,	/* 0 */	/* 0	/* 428 */			FC_END */		/*
/	0x0,	/ 0	FC_PP */	0x4b,	/*	/* 470 */		
/* 342 */ 0x0,	/* 0 */		FC_PAD */	0x5c,	/*	FC_BOGUS_ARRAY */		0x21,
70 */	0x46,	/*	/* 430 */					/* 3
/* 344 */			FC_VARIABLE_REPEAT */	0x48,	/*	/*		
[pointer_deref] */	0x12, 0x10, /* FC_UP		FC_FIXED_OFFSET */	0x49,	/*	/*		0x0,
/* 346 */ NdrFcShort(0x2),	/* Offset= 2 (348) */		/* 432 */ NdrFcShort(0x4),	/* 4 */		/*		/*
/* 348 */			/* 434 */ NdrFcShort(0x0),	/* 0 */		/* 476 */ NdrFcShort(0x0),	/* 0 */	
	0x12, 0x0, /* FC_UP */		/* 436 */ NdrFcShort(0x1),	/* 1 */		/* 478 */ NdrFcLong(0xffffffff),	/* -1 */	
/* 350 */ NdrFcShort(0x1fc),	/* Offset= 508 (858) */		/* 438 */ NdrFcShort(0x0),	/* 0 */		/* 482 */ 0x4c,	/* FC_EMBEDDED_COMPLEX	
/* 352 */			/* 440 */ NdrFcShort(0x0),	/* 0 */		/*		/* 0
FC_ENCAPSULATED_UNION */	0x2a,	/*	/* 442 */ 0x12, 0x0, /* FC_UP */	/* Offset= -146 (298) */		/*		0x0,
	0x49,	/*	/* 444 */ NdrFcShort(0xffffffff6e),			/* 484 */ NdrFcShort(0xfffff50),	/* Offset= -176 (308) */	/*
73 */			/* 446 */			/* 486 */ 0x5c,	/* FC_PAD */	/*
/* 354 */ NdrFcShort(0x18),	/* 24 */		FC_END */	0x5b,	/*			0x5b,
/* 356 */ NdrFcShort(0xa),	/* 10 */					FC_END */		/*
/* 358 */ NdrFcLong(0x8),	/* 8 */			0x8,	/*	/* 488 */		
/* 362 */ NdrFcShort(0x58),	/* Offset= 88 (450) */		FC_LONG */					0x1a,
/* 364 */ NdrFcLong(0xd),	/* 13 */		/* 448 */ 0x5c,	/* FC_PAD */		FC_BOGUS_STRUCT */		/* 3
/* 368 */ NdrFcShort(0x78),	/* Offset= 120 (488) */		FC_END */	0x5b,	/*	/*		0x3,
/* 370 */ NdrFcLong(0x9),	/* 9 */		/* 450 */			/*		
/* 374 */ NdrFcShort(0x94),	/* Offset= 148 (522) */			0x16,	/*	/* 490 */ NdrFcShort(0x8),	/* 8 */	
/* 376 */ NdrFcLong(0xc),	/* 12 */		FC_PSTRUCT */	0x3,	/* 3	/* 492 */ NdrFcShort(0x0),	/* 0 */	
/* 380 */ NdrFcShort(0xbc),	/* Offset= 188 (568) */		/*			/* 494 */ NdrFcShort(0x6),	/* Offset= 6 (500) */	
/* 382 */ NdrFcLong(0x24),	/* 36 */		/* 452 */ NdrFcShort(0x8),	/* 8 */		/* 496 */ 0x8,	/* FC_LONG */	/*
/* 386 */ NdrFcShort(0x114),	/* Offset= 276 (662) */		/* 454 */				0x36,	
/* 388 */ NdrFcLong(0x800d),	/* 32781 */		FC_PP */	0x4b,	/*	FC_POINTER */		
/* 392 */ NdrFcShort(0x130),	/* Offset= 304 (696) */			0x5c,	/*	/* 498 */ 0x5c,	/* FC_PAD */	/*
/* 394 */ NdrFcLong(0x10),	/* 16 */		FC_PAD */			/* 500 */		
/* 398 */ NdrFcShort(0x148),	/* Offset= 328 (726) */		/* 456 */					0x11, 0x0, /* FC_RP */
/* 400 */ NdrFcLong(0x2),	/* 2 */		FC_NO_REPEAT */	0x46,	/*	/* 502 */ NdrFcShort(0xfffffe0),	/* Offset= -32 (470) */	
/* 404 */ NdrFcShort(0x160),	/* Offset= 352 (756) */		FC_PAD */	0x5c,	/*	/* 504 */		0x21,
/* 406 */ NdrFcLong(0x3),	/* 3 */		/* 458 */ NdrFcShort(0x4),	/* 4 */		FC_BOGUS_ARRAY */		/* 3
/* 410 */ NdrFcShort(0x178),	/* Offset= 376 (786) */					/*		
/* 412 */ NdrFcLong(0x14),	/* 20 */							
/* 416 */ NdrFcShort(0x190),	/* Offset= 400 (816) */							
/* 418 */ NdrFcShort(0xffffffff),	/* Offset= -1 (417) */							

```

/* 506 */ NdrFcShort( 0x0 ), /* 0 */
/* 508 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */
0x0, /*
*/
/* 510 */ NdrFcShort( 0x0 ), /* 0 */
/* 512 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 516 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /* 0
*/
/* 518 */ NdrFcShort( 0xfffff40 ), /* Offset= -192 (326) */
/* 520 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 522 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3
*/
/* 524 */ NdrFcShort( 0x8 ), /* 8 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 530 */ 0x8, /* FC_LONG */
0x36, /*
FC_POINTER */
/* 532 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 534 */
0x11, 0x0, /* FC_RP */
/* 536 */ NdrFcShort( 0xfffffe0 ), /* Offset= -32 (504) */
/* 538 */
0x1b, /*
FC_CARRAY */
0x3, /* 3
*/
/* 540 */ NdrFcShort( 0x4 ), /* 4 */
/* 542 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */
0x0, /*
*/
/* 544 */ NdrFcShort( 0x0 ), /* 0 */
/* 546 */
0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */
/* 548 */
0x48, /*
FC_VARIABLE_REPEAT */

```

```

/* 550 */ NdrFcShort( 0x4 ), 0x49, /*
*/
/* 552 */ NdrFcShort( 0x0 ), /* 4 */
/* 554 */ NdrFcShort( 0x1 ), /* 0 */
/* 556 */ NdrFcShort( 0x0 ), /* 1 */
/* 558 */ NdrFcShort( 0x0 ), /* 0 */
/* 560 */ 0x12, 0x0, /* FC_UP */
/* 562 */ NdrFcShort( 0x182 ), /* Offset= 386 (948) */
/* 564 */
0x5b, /*
FC_END */
0x8, /*
FC_LONG */
/* 566 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 568 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3
*/
/* 570 */ NdrFcShort( 0x8 ), /* 8 */
/* 572 */ NdrFcShort( 0x0 ), /* 0 */
/* 574 */ NdrFcShort( 0x6 ), /* Offset= 6 (580) */
/* 576 */ 0x8, /* FC_LONG */
0x36, /*
FC_POINTER */
/* 578 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 580 */
0x11, 0x0, /* FC_RP */
/* 582 */ NdrFcShort( 0xfffffd4 ), /* Offset= -44 (538) */
/* 584 */
0x2f, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 586 */ NdrFcLong( 0x2f ), /* 47 */
/* 590 */ NdrFcShort( 0x0 ), /* 0 */
/* 592 */ NdrFcShort( 0x0 ), /* 0 */
/* 594 */ 0xc0, /* 192 */
0x0, /* 0
*/
/* 596 */ 0x0, /* 0 */
0x0, /* 0
*/
/* 598 */ 0x0, /* 0 */
0x0, /* 0
*/
/*
*/
/* 600 */ 0x0, /* 0 */

```

```

/* 602 */ 0x46, /*
*/
0x1b, /*
FC_CARRAY */
0x0, /* 0
*/
/* 604 */ NdrFcShort( 0x1 ), /* 1 */
/* 606 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */
0x0, /*
*/
/* 608 */ NdrFcShort( 0x4 ), /* 4 */
/* 610 */ 0x1, /* FC_BYTE */
0x5b, /*
FC_END */
/* 612 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3
*/
/* 614 */ NdrFcShort( 0x10 ), /* 16 */
/* 616 */ NdrFcShort( 0x0 ), /* 0 */
/* 618 */ NdrFcShort( 0xa ), /* Offset= 10 (628) */
/* 620 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 622 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /* 0
*/
/* 624 */ NdrFcShort( 0xfffffd8 ), /* Offset= -40 (584) */
/* 626 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 628 */
0x12, 0x0, /* FC_UP */
/* 630 */ NdrFcShort( 0xfffffe4 ), /* Offset= -28 (602) */
/* 632 */
0x1b, /*
FC_CARRAY */
0x3, /* 3
*/
/* 634 */ NdrFcShort( 0x4 ), /* 4 */
/* 636 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */
0x0, /*
*/
/* 638 */ NdrFcShort( 0x0 ), /* 0 */
/* 640 */
0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */

```

/* 642 */			/* 688 */ NdrFcShort(0x10),	/* 716 */ FC_LONG */	/* 730 */	0x4b,	/*
FC_VARIABLE_REPEAT */	0x48,	/*	FC_SHORT */	0x6,	/*	FC_PP */	
	0x49,	/*	/* 690 */ 0x6,	/* FC_SHORT */		0x5c,	/*
FC_FIXED_OFFSET */			FC_EMBEDDED_COMPLEX */	0x4c,	/*	FC_PAD */	
/* 644 */ NdrFcShort(0x4),	/* 4 */		/* 692 */ 0x0,	/* 0 */		/* 732 */	
/* 646 */ NdrFcShort(0x0),	/* 0 */		Offset= -15 (678) */	NdrFcShort(0xffffffff),	/*	FC_NO_REPEAT */	
/* 648 */ NdrFcShort(0x1),	/* 1 */		FC_END */	0x5b,	/*	0x46,	/*
/* 650 */ NdrFcShort(0x0),	/* 0 */		/* 696 */			0x5c,	/*
/* 652 */ NdrFcShort(0x0),	/* 0 */		FC_BOGUS_STRUCT */	0x1a,	/*	FC_PAD */	
/* 654 */ 0x12, 0x0, /* FC_UP */				0x3,	/* 3	/* 734 */ NdrFcShort(0x4),	/* 4 */
/* 656 */ NdrFcShort(0xffffffffd4),	/* Offset= -44 (612) */					/* 736 */ NdrFcShort(0x4),	/* 4 */
/* 658 */						/* 738 */ 0x12, 0x0, /* FC_UP */	
	0x5b,	/*				/* 740 */ NdrFcShort(0xffffffffe8),	/* Offset= -24 (716) */
FC_END */						/* 742 */	
	0x8,	/*				0x5b,	/*
FC_LONG */			/* 698 */ NdrFcShort(0x18),	/* 24 */		FC_END */	
/* 660 */ 0x5c,	/* FC_PAD */		/* 700 */ NdrFcShort(0x0),	/* 0 */		0x8,	/*
	0x5b,	/*	/* 702 */ NdrFcShort(0xa),	/* Offset= 10 (712) */		FC_LONG */	
FC_END */			/* 704 */ 0x8,	/* FC_LONG */		/* 744 */ 0x8,	/* FC_LONG */
/* 662 */			FC_POINTER */	0x36,	/*	0x5b,	/*
	0x1a,	/*	/* 706 */ 0x4c,	/* FC_EMBEDDED_COMPLEX		FC_END */	
FC_BOGUS_STRUCT */			*/	0x0,	/* 0	/* 746 */	
	0x3,	/* 3				FC_CARRAY */	
/						0x1b,	/
/* 664 */ NdrFcShort(0x8),	/* 8 */		/* 708 */ NdrFcShort(0xffffffffe8),	/* Offset= -24 (684) */		0x1,	/* 1
/* 666 */ NdrFcShort(0x0),	/* 0 */		/* 710 */ 0x5c,	/* FC_PAD */			
/* 668 */ NdrFcShort(0x6),	/* Offset= 6 (674) */			0x5b,	/*	/* 748 */ NdrFcShort(0x2),	/* 2 */
/* 670 */ 0x8,	/* FC_LONG */		FC_END */			/* 750 */ 0x19,	/* Corr desc: field pointer,
	0x36,	/*	/* 712 */			FC_ULONG */	
FC_POINTER */				0x11, 0x0, /* FC_RP */		0x0,	/*
/* 672 */ 0x5c,	/* FC_PAD */		/* 714 */ NdrFcShort(0xffffffff0c),	/* Offset= -244 (470) */		/* 752 */ NdrFcShort(0x0),	/* 0 */
	0x5b,	/*	/* 716 */			/* 754 */ 0x6,	/* FC_SHORT */
FC_END */				0x1b,	/*	0x5b,	/*
/* 674 */			FC_CARRAY */	0x0,	/* 0	FC_END */	
	0x11, 0x0, /* FC_RP */					/* 756 */	
/* 676 */ NdrFcShort(0xffffffffd4),	/* Offset= -44 (632) */					0x16,	/*
/* 678 */			/* 718 */ NdrFcShort(0x1),	/* 1 */		FC_PSTRUCT */	
	0x1d,	/*	/* 720 */ 0x19,	/* Corr desc: field pointer,		0x3,	/* 3
FC_SMFARRAY */			FC_ULONG */				
	0x0,	/* 0		0x0,	/*	/* 758 */ NdrFcShort(0x8),	/* 8 */
/						/ 760 */	
/* 680 */ NdrFcShort(0x8),	/* 8 */		/* 722 */ NdrFcShort(0x0),	/* 0 */		0x4b,	/*
/* 682 */ 0x2,	/* FC_CHAR */		/* 724 */ 0x1,	/* FC_BYTE */		0x5c,	/*
	0x5b,	/*	FC_END */	0x5b,	/*	FC_PP */	
FC_END */			/* 726 */			FC_PAD */	
/* 684 */				0x16,	/*	/* 762 */	
FC_STRUCT */	0x15,	/*	FC_PSTRUCT */	0x3,	/* 3	0x46,	/*
	0x3,	/* 3				FC_NO_REPEAT */	
/						0x5c,	/
			/* 728 */ NdrFcShort(0x8),	/* 8 */		FC_PAD */	
						/* 764 */ NdrFcShort(0x4),	/* 4 */

```

/* 766 */ NdrFcShort( 0x4 ), /* 4 */
/* 768 */ 0x12, 0x0, /* FC_UP */
/* 770 */ NdrFcShort( 0xfffffe8 ), /* Offset= -24 (746) */
/* 772 */
FC_END */
0x5b, /*
FC_LONG */
/* 774 */ 0x8, /* FC_LONG */
/* 776 */ 0x5b, /*
FC_END */
/* 776 */
FC_CARRY */
0x1b, /*
FC_CARRY */
0x3, /* 3
*/
/* 778 */ NdrFcShort( 0x4 ), /* 4 */
/* 780 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */
0x0, /*
*/
/* 782 */ NdrFcShort( 0x0 ), /* 0 */
/* 784 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */
/* 786 */
0x16, /*
FC_PSTRUCT */
0x3, /* 3
*/
/* 788 */ NdrFcShort( 0x8 ), /* 8 */
/* 790 */
0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */
/* 792 */
0x46, /*
FC_NO_REPEAT */
0x5c, /*
FC_PAD */
/* 794 */ NdrFcShort( 0x4 ), /* 4 */
/* 796 */ NdrFcShort( 0x4 ), /* 4 */
/* 798 */ 0x12, 0x0, /* FC_UP */
/* 800 */ NdrFcShort( 0xfffffe8 ), /* Offset= -24 (776) */
/* 802 */
FC_END */
0x5b, /*
FC_LONG */
0x8, /*

```

```

/* 804 */ 0x8,
FC_END */
/* 806 */
FC_CARRY */
0x1b, /*
0x7, /* 7
*/
/* 808 */ NdrFcShort( 0x8 ), /* 8 */
/* 810 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */
0x0, /*
*/
/* 812 */ NdrFcShort( 0x0 ), /* 0 */
/* 814 */ 0xb, /* FC_HYPER */
0x5b, /*
FC_END */
/* 816 */
0x16, /*
FC_PSTRUCT */
0x3, /* 3
*/
/* 818 */ NdrFcShort( 0x8 ), /* 8 */
/* 820 */
0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */
/* 822 */
0x46, /*
FC_NO_REPEAT */
0x5c, /*
FC_PAD */
/* 824 */ NdrFcShort( 0x4 ), /* 4 */
/* 826 */ NdrFcShort( 0x4 ), /* 4 */
/* 828 */ 0x12, 0x0, /* FC_UP */
/* 830 */ NdrFcShort( 0xfffffe8 ), /* Offset= -24 (806) */
/* 832 */
0x5b, /*
FC_END */
0x8, /*
FC_LONG */
/* 834 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */
/* 836 */
0x15, /*
0x3, /* 3
*/
*/
/* 838 */ NdrFcShort( 0x8 ), /* 8 */
/* 840 */ 0x8, /* FC_LONG */

```

```

FC_END */
0x5c, /*
FC_END */
/* 844 */
FC_CARRY */
0x1b, /*
0x3, /* 3
*/
/* 846 */ NdrFcShort( 0x8 ), /* 8 */
/* 848 */ 0x7, /* Corr desc: FC_USHORT */
0x0, /*
*/
/* 850 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 852 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /* 0
*/
/* 854 */ NdrFcShort( 0xfffffee ), /* Offset= -18 (836) */
/* 856 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 858 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3
*/
/* 860 */ NdrFcShort( 0x28 ), /* 40 */
/* 862 */ NdrFcShort( 0xfffffee ), /* Offset= -18 (844) */
/* 864 */ NdrFcShort( 0x0 ), /* Offset= 0 (864) */
/* 866 */ 0x6, /* FC_SHORT */
0x6, /*
FC_SHORT */
/* 868 */ 0x38, /* FC_ALIGNM4 */
0x8, /*
FC_LONG */
/* 870 */ 0x8, /* FC_LONG */
0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 872 */ 0x0, /* 0 */
NdrFcShort( 0xfffffd7 ), /*
Offset= -521 (352) */
0x5b, /*
FC_END */
/* 876 */
0x12, 0x0, /* FC_UP */
/* 878 */ NdrFcShort( 0xfffffe6 ), /* Offset= -266 (612) */
/* 880 */
0x12, 0x8, /* FC_UP
[simple_pointer] */
/* 882 */ 0x1, /* FC_BYTE */
0x5c, /*
FC_PAD */

```

```

/* 884 */
[pointer_deref] */
/* 886 */ 0x6,
FC_PAD */
/* 888 */
[pointer_deref] */
/* 890 */ 0x8,
FC_PAD */
/* 892 */
[pointer_deref] */
/* 894 */ 0xa,
FC_PAD */
/* 896 */
[pointer_deref] */
/* 898 */ 0xc,
FC_PAD */
/* 900 */
/* 902 */ NdrFcShort( 0xffffd90 ), /* Offset= -624 (278) */
/* 904 */
[pointer_deref] */
/* 906 */ NdrFcShort( 0xffffd92 ), /* Offset= -622 (284) */
/* 908 */
[pointer_deref] */
/* 910 */ NdrFcShort( 0xffffda6 ), /* Offset= -602 (308) */
/* 912 */
[pointer_deref] */
/* 914 */ NdrFcShort( 0xffffdb4 ), /* Offset= -588 (326) */
/* 916 */
[pointer_deref] */
/* 918 */ NdrFcShort( 0xffffdc2 ), /* Offset= -574 (344) */
/* 920 */
[pointer_deref] */
/* 922 */ NdrFcShort( 0x2 ), /* Offset= 2 (924) */
/* 924 */
/* 926 */ NdrFcShort( 0x16 ), /* Offset= 22 (948) */
/* 928 */

```

```

0x12, 0x8, /* FC_UP
/* FC_SHORT */
0x5c,
0x12, 0x8, /* FC_UP
/* FC_LONG */
0x5c,
0x12, 0x8, /* FC_UP
/* FC_FLOAT */
0x5c,
0x12, 0x8, /* FC_UP
/* FC_DOUBLE */
0x5c,
0x12, 0x0, /* FC_UP */
/* Offset= -624 (278) */
0x12, 0x10, /* FC_UP
/* Offset= -622 (284) */
0x12, 0x10, /* FC_UP
/* Offset= -602 (308) */
0x12, 0x10, /* FC_UP
/* Offset= -588 (326) */
0x12, 0x10, /* FC_UP
/* Offset= -574 (344) */
0x12, 0x10, /* FC_UP
/* Offset= 2 (924) */
0x12, 0x0, /* FC_UP */
/* Offset= 22 (948) */

```

```

FC_STRUCT */
/*
/* 930 */ NdrFcShort( 0x10 ),
/* 932 */ 0x6,
FC_BYTE */
/* 934 */ 0x1,
FC_ALIGNM4 */
/* 936 */ 0x8,
FC_ALIGNM8 */
/* 938 */ 0xb,
FC_END */
/* 940 */
/* 942 */ NdrFcShort( 0xfffff2 ), /* Offset= -14 (928) */
/* 944 */
[pointer_deref] */
/* 946 */ 0x2,
FC_PAD */
/* 948 */
FC_BOGUS_STRUCT */
/*
/* 950 */ NdrFcShort( 0x20 ),
/* 952 */ NdrFcShort( 0x0 ),
/* 954 */ NdrFcShort( 0x0 ),
/* 956 */ 0x8,
FC_LONG */
/* 958 */ 0x6,
FC_SHORT */
/* 960 */ 0x6,
FC_SHORT */
/* 962 */ 0x4c,
/*
/* 964 */ NdrFcShort( 0xffffc42 ), /* Offset= -958 (6) */
/* 966 */ 0x5c,
0x5b,
FC_END */
/* 968 */ 0xb4,
131 */

```

```

0x15,
0x7,
/* 16 */
/* FC_SHORT */
0x1,
/* FC_BYTE */
0x38,
/* FC_LONG */
0x39,
/* FC_HYPER */
0x5b,
0x12, 0x0, /* FC_UP */
/* Offset= -14 (928) */
0x12, 0x8, /* FC_UP
/* FC_CHAR */
0x5c,
0x1a,
0x7,
/* 32 */
/* 0 */
/* Offset= 0 (954) */
/* FC_LONG */
0x8,
/* FC_SHORT */
0x6,
/* FC_SHORT */
0x6,
/* FC_EMBEDDED_COMPLEX
0x0,
/* 0
/* Offset= -958 (6) */
/* FC_PAD */
0x5b,
/* FC_USER_MARSHAL */
0x83,

```

```

/* 970 */ NdrFcShort( 0x00 ), /* 0 */
/* 974 */ NdrFcShort( 0x0 ), /* 0 */
/* 976 */ NdrFcShort( 0xffffc32 ), /* Offset= -974 (2) */
/* 978 */
0x11, 0x4, /* FC_RP
/* Offset= (986) */
/* 982 */
0x13, 0x0, /* FC_OP */
/* 984 */ NdrFcShort( 0xfffffdc ), /* Offset= -36 (948) */
/* 986 */ 0xb4, /* FC_USER_MARSHAL */
0x83,
131 */
/* 988 */ NdrFcShort( 0x0 ), /* 0 */
/* 990 */ NdrFcShort( 0x10 ), /* 16 */
/* 992 */ NdrFcShort( 0x0 ), /* 0 */
/* 994 */ NdrFcShort( 0xfffffff4 ), /* Offset= -12 (982) */
0x0
}
};
const CInterfaceProxyVtbl * _tpcc_com_ps_ProxyVtblList[] =
{
( CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
0
};
const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[] =
{
( CInterfaceStubVtbl *) &_ITPCCStubVtbl,
0
};
PCInterfaceName const _tpcc_com_ps_InterfaceNamesList[] =
{
"ITPCC",
0
};
#define _tpcc_com_ps_CHECK_IID(n) IID_GENERIC_CHECK_IID(
_tpcc_com_ps, pIID, n)
int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex
)
{
if(!_tpcc_com_ps_CHECK_IID(0))
{
*pIndex = 0;
return 1;
}
}

```

```

return 0;
}

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

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

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

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

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

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

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

```

```

#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,{0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00}} */

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

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

#pragma code_seg(".orpc")

```

```

static const unsigned short ITPCC_FormatStringOffsetTable[] =
{
    0,
    44,
    88,
    132,
    176,
    220
};

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

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

```



```

NdrFcShort( 0x8 ), /*
axp64 Stack size/offset = 8 */
#endif
/* 74 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */
/* Parameter txn_out */

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

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

/* Procedure Delivery */

/* 88 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 90 */ NdrFcLong( 0x0 ), /* 0 */
/* 94 */ NdrFcShort( 0x5 ), /* 5 */
#ifndef _ALPHA_
/* 96 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56
*/
#else
NdrFcShort( 0x30 ), /*
axp64 Stack size/offset = 48 */
#endif
/* 98 */ NdrFcShort( 0x0 ), /* 0 */
/* 100 */ NdrFcShort( 0x8 ), /* 8 */
/* 102 */ 0x47, /* Oi2 Flags: srv must size, clt
must size, has return, has ext, */

```

```

0x3, /* 3
*/ 104 */ 0xa, /* 10 */
0x7, /*
Ext Flags: new corr desc, clt corr check, srv corr check, */
/* 106 */ NdrFcShort( 0x20 ), /* 32 */
/* 108 */ NdrFcShort( 0x20 ), /* 32 */
/* 110 */ NdrFcShort( 0x0 ), /* 0 */
/* 112 */ NdrFcShort( 0x0 ), /* 0 */
/* Parameter txn_in */

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

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

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

/* Procedure StockLevel */

/* 132 */ 0x33, /* FC_AUTO_HANDLE */

```

```

0x6c, /*
Old Flags: object, Oi2 */
/* 138 */ NdrFcShort( 0x6 ), /* 6 */
#ifndef _ALPHA_
/* 140 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56
*/
#else
NdrFcShort( 0x30 ), /*
axp64 Stack size/offset = 48 */
#endif
/* 142 */ NdrFcShort( 0x0 ), /* 0 */
/* 144 */ NdrFcShort( 0x8 ), /* 8 */
/* 146 */ 0x47, /* Oi2 Flags: srv must size, clt
must size, has return, has ext, */
0x3, /* 3
*/
/* 148 */ 0xa, /* 10 */
0x7, /*
Ext Flags: new corr desc, clt corr check, srv corr check, */
/* 150 */ NdrFcShort( 0x20 ), /* 32 */
/* 152 */ NdrFcShort( 0x20 ), /* 32 */
/* 154 */ NdrFcShort( 0x0 ), /* 0 */
/* 156 */ NdrFcShort( 0x0 ), /* 0 */
/* Parameter txn_in */

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

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

```

```

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

/* Procedure OrderStatus */

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

/* Parameter txn_in */

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

```

```

/* 206 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */
/* Parameter txn_out */

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

/* Return value */

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

/* Procedure CallSetComplete */

/* 220 */ 0x33, /* FC_AUTO_HANDLE */
/* 0x6c, */
Old Flags: object, Oi2 */
/* 222 */ NdrFcLong( 0x0 ), /* 0 */
/* 226 */ NdrFcShort( 0x8 ), /* 8 */
/* 228 */ NdrFcShort( 0x10 ), /* ia64, axp64 Stack size/offset
= 16 */
/* 230 */ NdrFcShort( 0x0 ), /* 0 */
/* 232 */ NdrFcShort( 0x8 ), /* 8 */
/* 234 */ 0x44, /* Oi2 Flags: has return, has
ext, */
/* 0x1, */
/* 1

/* 236 */ 0xa, /* 10 */
/* 0x1, */

Ext Flags: new corr desc, */
/* 238 */ NdrFcShort( 0x0 ), /* 0 */
/* 240 */ NdrFcShort( 0x0 ), /* 0 */
/* 242 */ NdrFcShort( 0x0 ), /* 0 */
/* 244 */ NdrFcShort( 0x0 ), /* 0 */

/* Return value */

```

```

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

}
};

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
0,
{
/*
/* 2 */
/* 4 */ NdrFcShort( 0x39e ), /* 0x12, 0x0, */
/* 6 */ /* FC_UP */
/* Offset= 926 (930) */
/* 0x2b, */
FC_NON_ENCAPSULATED_UNION */
/* 0x9, */
FC_ULONG */
/* 8 */ 0x7, /* Corr desc: FC_USHORT */
/* 0x0, */
/*
/* 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 */ NdrFcLong( 0xfffff ), /* Offset= -1 (277) */
/* 280 */

FC_STRUCT */
0x15, /*
0x7, /* 7
*/
/* 282 */ NdrFcShort( 0x8 ), /* 8 */
/* 284 */ 0xb, /* FC_HYPER */
0x5b, /*
FC_END */
/* 286 */
0x12, 0x0, /* FC_UP */
/* 288 */ NdrFcShort( 0xe ), /* Offset= 14 (302) */
/* 290 */
0x1b, /*
0x1, /* 1
*/
*/
/* 292 */ NdrFcShort( 0x2 ), /* 2 */
/* 294 */ 0x9, /* Corr desc: FC_ULONG */
0x0, /*
*/
/* 296 */ NdrFcShort( 0xffc ), /* -4 */
/* 298 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 300 */ 0x6, /* FC_SHORT */

```

```

0x5b, /*
0x17, /*
FC_CSTRUCT */
0x3, /* 3
*/
/* 304 */ NdrFcShort( 0x8 ), /* 8 */
/* 306 */ NdrFcShort( 0xfffff0 ), /* Offset= -16 (290) */
/* 308 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 310 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 312 */
0x2f, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 314 */ NdrFcLong( 0x0 ), /* 0 */
/* 318 */ NdrFcShort( 0x0 ), /* 0 */
/* 320 */ NdrFcShort( 0x0 ), /* 0 */
/* 322 */ 0xc0, /* 192 */
0x0, /* 0
*/
/* 324 */ 0x0, /* 0
*/
/* 326 */ 0x0, /* 0
*/
0x0, /* 0
*/
/* 328 */ 0x0, /* 0
*/
70 */
/* 330 */
0x2f, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 332 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 336 */ NdrFcShort( 0x0 ), /* 0 */
/* 338 */ NdrFcShort( 0x0 ), /* 0 */
/* 340 */ 0xc0, /* 192 */
0x0, /* 0
*/
/* 342 */ 0x0, /* 0
*/
0x0, /* 0
*/
/* 344 */ 0x0, /* 0
*/
0x0, /* 0
*/
/* 346 */ 0x0, /* 0
*/

```

```

0x46, /*
70 */
/* 348 */

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

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

FC_ENCAPSULATED_UNION */

0x2a, /*
0x89, /*

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

0x21, /*
0x3, /* 3

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

0x0, /*

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

```

```

/* 442 */ NdrFcShort( 0xfffff74 ), /* Offset= -140 (202) */
/* 444 */ 0x5c, /* FC_PAD */

0x5b, /*

FC_END */
/* 446 */

0x1a, /*
0x3, /* 3

*/
/* 448 */ NdrFcShort( 0x10 ), /* 16 */
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */ NdrFcShort( 0x6 ), /* Offset= 6 (458) */
/* 454 */ 0x8, /* FC_LONG */

0x39, /*
0x36, /* FC_POINTER */

0x5b, /*

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

0x21, /*
0x3, /* 3

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

0x0, /*

*/
/* 468 */ NdrFcShort( 0x0 ), /* 0 */
/* 470 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 472 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 476 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 478 */ 0x4c, /* FC_EMBEDDED_COMPLEX

0x0, /* 0

*/
/* 480 */ NdrFcShort( 0xfffff58 ), /* Offset= -168 (312) */
/* 482 */ 0x5c, /* FC_PAD */

0x5b, /*

FC_END */
/* 484 */

0x1a, /*
0x3, /* 3

*/
/* 486 */ NdrFcShort( 0x10 ), /* 16 */
/* 488 */ NdrFcShort( 0x0 ), /* 0 */
/* 490 */ NdrFcShort( 0x6 ), /* Offset= 6 (496) */
/* 492 */ 0x8, /* FC_LONG */

```

```

FC_ALIGNM8 0x36, 0x39, /*
/* FC_POINTER */ /*
0x5b, /*

FC_END */
/* 496 */

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

0x21, /*
0x3, /* 3

*/
/* 502 */ NdrFcShort( 0x0 ), /* 0 */
/* 504 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */

0x0, /*

*/
/* 506 */ NdrFcShort( 0x0 ), /* 0 */
/* 508 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 510 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 514 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 516 */ 0x4c, /* FC_EMBEDDED_COMPLEX

0x0, /* 0

*/
/* 518 */ NdrFcShort( 0xfffff44 ), /* Offset= -188 (330) */
/* 520 */ 0x5c, /* FC_PAD */

0x5b, /*

FC_END */
/* 522 */

0x1a, /*
0x3, /* 3

*/
/* 524 */ NdrFcShort( 0x10 ), /* 16 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 530 */ 0x8, /* FC_LONG */

0x39, /*
0x36, /* FC_POINTER */

0x5b, /*

FC_END */
/* 534 */

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

0x21, /*
0x3, /* 3

*/
/* 540 */ NdrFcShort( 0x0 ), /* 0 */

```

```

/* 542 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */
0x0, /*
*/
/* 544 */ NdrFcShort( 0x0 ), /* 0 */
/* 546 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 548 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 552 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 554 */
0x12, 0x0, /* FC_UP */
/* 556 */ NdrFcShort( 0x176 ), /* Offset= 374 (930) */
/* 558 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 560 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3
*/
/* 562 */ NdrFcShort( 0x10 ), /* 16 */
/* 564 */ NdrFcShort( 0x0 ), /* 0 */
/* 566 */ NdrFcShort( 0x6 ), /* Offset= 6 (572) */
/* 568 */ 0x8, /* FC_LONG */
0x39, /*
FC_ALIGNM8 */
/* 570 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 572 */
0x11, 0x0, /* FC_RP */
/* 574 */ NdrFcShort( 0xffffffffc ), /* Offset= -36 (538) */
/* 576 */
0x2f, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 578 */ NdrFcLong( 0x2f ), /* 47 */
/* 582 */ NdrFcShort( 0x0 ), /* 0 */
/* 584 */ NdrFcShort( 0x0 ), /* 0 */
/* 586 */ 0xc0, /* 192 */
0x0, /* 0
*/
/* 588 */ 0x0, /* 0 */
0x0, /* 0
*/
/* 590 */ 0x0, /* 0 */
0x0, /* 0
*/
/* 592 */ 0x0, /* 0 */
0x46, /*
70 */
/* 594 */

```

```

FC_CARRAY */
0x1b, /*
0x0, /* 0
*/
/* 596 */ NdrFcShort( 0x1 ), /* 1 */
/* 598 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */
0x0, /*
*/
/* 600 */ NdrFcShort( 0x4 ), /* 4 */
/* 602 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 604 */ 0x1, /* FC_BYTE */
0x5b, /*
FC_END */
/* 606 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3
*/
/* 608 */ NdrFcShort( 0x18 ), /* 24 */
/* 610 */ NdrFcShort( 0x0 ), /* 0 */
/* 612 */ NdrFcShort( 0xc ), /* Offset= 12 (624) */
/* 614 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 616 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /* 0
*/
/* 618 */ NdrFcShort( 0xffffffffd6 ), /* Offset= -42 (576) */
/* 620 */ 0x39, /* FC_ALIGNM8 */
0x36, /*
FC_POINTER */
/* 622 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 624 */
0x12, 0x0, /* FC_UP */
/* 626 */ NdrFcShort( 0xffffffffe0 ), /* Offset= -32 (594) */
/* 628 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /* 3
*/
/* 630 */ NdrFcShort( 0x0 ), /* 0 */
/* 632 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */
0x0, /*
*/
/* 634 */ NdrFcShort( 0x0 ), /* 0 */
/* 636 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 638 */ NdrFcLong( 0xfffffffff ), /* -1 */
/* 642 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 644 */

```

```

/* 646 */ NdrFcShort( 0xffffffffd8 ), /* Offset= -40 (608) */
/* 648 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 650 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3
*/
/* 652 */ NdrFcShort( 0x10 ), /* 16 */
/* 654 */ NdrFcShort( 0x0 ), /* 0 */
/* 656 */ NdrFcShort( 0x6 ), /* Offset= 6 (662) */
/* 658 */ 0x8, /* FC_LONG */
0x39, /*
FC_ALIGNM8 */
/* 660 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 662 */
0x11, 0x0, /* FC_RP */
/* 664 */ NdrFcShort( 0xffffffffdc ), /* Offset= -36 (628) */
/* 666 */
0x1d, /*
FC_SMFARRAY */
0x0, /* 0
*/
/* 668 */ NdrFcShort( 0x8 ), /* 8 */
/* 670 */ 0x2, /* FC_CHAR */
0x5b, /*
FC_END */
/* 672 */
0x15, /*
FC_STRUCT */
0x3, /* 3
*/
/* 674 */ NdrFcShort( 0x10 ), /* 16 */
/* 676 */ 0x8, /* FC_LONG */
0x6, /*
FC_SHORT */
/* 678 */ 0x6, /* FC_SHORT */
0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 680 */ 0x0, /* 0 */
NdrFcShort( 0xfffffffff1 ), /*
Offset= -15 (666) */
0x5b, /*
FC_END */
/* 684 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3
*/
/* 686 */ NdrFcShort( 0x20 ), /* 32 */

```

```

/* 688 */ NdrFcShort( 0x0 ), /* 0 */
/* 690 */ NdrFcShort( 0xa ), /* Offset= 10 (700) */
/* 692 */ 0x8, /* FC_LONG */
/* 694 */ 0x36, /* FC_POINTER */
/* 696 */ 0x0, /* 0 */
/* 698 */ 0x5b, /* FC_END */
/* 700 */
/* 702 */ NdrFcShort( 0xfffff10 ), /* 0x11, 0x0, /* FC_UP */
/* 704 */ /* Offset= -240 (462) */
/* 706 */ 0x1b, /* 0 */
/* 708 */ 0x0, /* 1 */
/* 710 */ 0x0, /* Corr desc: field pointer,
/* 712 */ 0x0, /* FC_BYTE */
/* 714 */ 0x5b, /* FC_END */
/* 716 */
/* 718 */ 0x1a, /* 0 */
/* 720 */ 0x3, /* 3 */
/* 722 */ NdrFcShort( 0x10 ), /* 16 */
/* 724 */ NdrFcShort( 0x0 ), /* 0 */
/* 726 */ NdrFcShort( 0x6 ), /* Offset= 6 (728) */
/* 728 */ 0x8, /* FC_LONG */
/* 730 */ 0x36, /* FC_POINTER */
/* 732 */ 0x5b, /* FC_END */
/* 734 */
/* 736 */ 0x12, 0x0, /* FC_UP */
/* 738 */ NdrFcShort( 0xfffffe6 ), /* Offset= -26 (704) */
/* 740 */ 0x1b, /* 0 */
/* 742 */
/* 744 */

```

```

/* 734 */ NdrFcShort( 0x2 ), /* 0x1, /* 1
/* 736 */ 0x19, /* 2 */
/* 738 */ FC_ULONG */ /* Corr desc: field pointer,
/* 740 */ 0x0, /* 0
/* 742 */ 0x0, /* 0
/* 744 */ 0x5b, /* FC_SHORT
/* 746 */ 0x1a, /* 1
/* 748 */ 0x3, /* 3
/* 750 */ NdrFcShort( 0x10 ), /* 16
/* 752 */ NdrFcShort( 0x0 ), /* 0
/* 754 */ NdrFcShort( 0x6 ), /* Offset= 6 (756)
/* 756 */ 0x8, /* FC_LONG
/* 758 */ 0x36, /* FC_POINTER
/* 760 */ 0x5b, /* FC_END
/* 762 */
/* 764 */ NdrFcShort( 0xfffffe6 ), /* 0x12, 0x0, /* FC_UP
/* 766 */ /* Offset= -26 (732)
/* 768 */ 0x1b, /* 0
/* 770 */ 0x3, /* 3
/* 772 */ NdrFcShort( 0x4 ), /* 4
/* 774 */ 0x19, /* Corr desc: field pointer,
/* 776 */ FC_ULONG */
/* 778 */ 0x0, /* 0
/* 780 */ NdrFcShort( 0x0 ), /* 0
/* 782 */ NdrFcShort( 0x1 ), /* Corr flags: early,
/* 784 */ 0x8, /* FC_LONG
/* 786 */ 0x5b, /* FC_END
/* 788 */
/* 790 */ NdrFcShort( 0x10 ), /* 16
/* 792 */ NdrFcShort( 0x0 ), /* 0
/* 794 */ NdrFcShort( 0x6 ), /* Offset= 6 (812)
/* 796 */ 0x8, /* FC_LONG
/* 798 */ 0x36, /* FC_POINTER
/* 800 */ 0x5b, /* FC_END
/* 802 */
/* 804 */ NdrFcShort( 0x12, 0x0, /* FC_UP
/* 806 */ NdrFcShort( 0xfffffe6 ), /* Offset= -26 (788)
/* 808 */ 0x15, /* 15
/* 810 */ 0x3, /* 3
/* 812 */ NdrFcShort( 0x8 ), /* 8
/* 814 */ 0x8, /* FC_LONG
/* 816 */ 0x8, /* FC_PAD
/* 818 */ 0x5c, /* FC_END
/* 820 */

```

```

/* 720 */ NdrFcShort( 0x36, /* 0x39, /*
/* 722 */ 0x5b, /* FC_POINTER
/* 724 */
/* 726 */ FC_END
/* 728 */
/* 730 */ 0x12, 0x0, /* FC_UP
/* 732 */ NdrFcShort( 0xfffffe6 ), /* Offset= -26 (760)
/* 734 */ 0x1b, /* 1b
/* 736 */ FC_CARRAY
/* 738 */
/* 740 */ 0x7, /* 7
/* 742 */ NdrFcShort( 0x8 ), /* 8
/* 744 */ 0x19, /* Corr desc: field pointer,
/* 746 */ FC_ULONG
/* 748 */ 0x0, /* 0
/* 750 */ NdrFcShort( 0x0 ), /* 0
/* 752 */ NdrFcShort( 0x1 ), /* Corr flags: early,
/* 754 */ 0xb, /* FC_HYPER
/* 756 */ 0x5b, /* FC_END
/* 758 */ 0x800
/* 760 */
/* 762 */ FC_BOGUS_STRUCT
/* 764 */ 0x1a, /* 1a
/* 766 */ 0x3, /* 3
/* 768 */ NdrFcShort( 0x10 ), /* 16
/* 770 */ NdrFcShort( 0x0 ), /* 0
/* 772 */ NdrFcShort( 0x6 ), /* Offset= 6 (812)
/* 774 */ 0x8, /* FC_LONG
/* 776 */ 0x39, /* FC_ALIGNM8
/* 778 */ 0x10, /* 10
/* 780 */ 0x36, /* FC_POINTER
/* 782 */ 0x5b, /* FC_END
/* 784 */ 0x12, 0x0, /* FC_UP
/* 786 */ NdrFcShort( 0xfffffe6 ), /* Offset= -26 (788)
/* 788 */ 0x15, /* 15
/* 790 */ 0x3, /* 3
/* 792 */ NdrFcShort( 0x8 ), /* 8
/* 794 */ 0x8, /* FC_LONG
/* 796 */ 0x8, /* FC_PAD
/* 798 */ 0x5c, /* FC_END
/* 800 */

```

FC_CARRAY */	0x1b,	/*	/* 868 */ 0x6,	FC_SHORT */	/*	FC_SHORT */ 0x1,	/*
/	0x3,	/ 3	FC_PAD */	/* 870 */		/* FC_BYTE */	/*
/* 826 */ NdrFcShort(0x8),	/* 8 */		[simple_pointer] */		0x12, 0x8, /* FC_UP	FC_ALIGNM4 */	
/* 828 */ 0x7,	/* Corr desc: FC_USHORT */		/* 872 */ 0x8,	/* FC_LONG */	0x5c,	/* 918 */ 0x8,	/* FC_LONG */
/	0x0,	/	FC_PAD */	/* 874 */		FC_ALIGNM8 */	/*
/* 830 */ NdrFcShort(0xffc8),	/* -56 */		/* 876 */ 0xa,	/* FC_FLOAT */	0x12, 0x8, /* FC_UP	/* 920 */ 0xb,	/* FC_HYPER */
/* 832 */ NdrFcShort(0x1),	/* Corr flags: early, */		[simple_pointer] */		0x5c,	FC_END */	/*
/* 834 */ 0x4c,	/* FC_EMBEDDED_COMPLEX		/* 878 */	/* FC_DOUBLE */	0x12, 0x8, /* FC_UP	/* 922 */	
/	0x0,	/ 0	FC_PAD */	/* 880 */ 0xc,	/* FC_DOUBLE */	/* 924 */ NdrFcShort(0xffffffff2),	0x12, 0x0, /* FC_UP */
/* 836 */ NdrFcShort(0xffffffffec),	/* Offset= -20 (816) */		/* 882 */	/* FC_DOUBLE */	0x5c,	/* 926 */	/* Offset= -14 (910) */
/* 838 */ 0x5c,	/* FC_PAD */	/*	[simple_pointer] */		0x12, 0x8, /* FC_UP	[simple_pointer] */	0x12, 0x8, /* FC_UP
FC_END */	0x5b,	/*	/* 884 */ 0xc,	/* FC_DOUBLE */	0x5c,	/* 928 */ 0x2,	/* FC_CHAR */
/* 840 */			FC_PAD */	/* 886 */		FC_PAD */	/*
FC_BOGUS_STRUCT */	0x1a,	/*	/* 888 */	/* FC_UP */	0x12, 0x0, /* FC_UP */	/* 930 */	/*
/	0x3,	/ 3	/* 888 */ NdrFcShort(0xffffffffda4),	/* Offset= -604 (280) */	/* Offset= -604 (280) */	FC_BOGUS_STRUCT */	0x1a,
/* 842 */ NdrFcShort(0x38),	/* 56 */		/* 886 */	/* FC_UP	0x12, 0x10, /* FC_UP	0x7,	/* 7
/* 844 */ NdrFcShort(0xffffffffec),	/* Offset= -20 (824) */		[pointer_deref] */		0x12, 0x10, /* FC_UP	*/	
/* 846 */ NdrFcShort(0x0),	/* Offset= 0 (846) */		/* 888 */ NdrFcShort(0xffffffffda6),	/* Offset= -602 (286) */	0x12, 0x10, /* FC_UP	/* 932 */ NdrFcShort(0x20),	/* 32 */
/* 848 */ 0x6,	/* FC_SHORT */	/*	/* 890 */	/* FC_UP	0x12, 0x10, /* FC_UP	/* 934 */ NdrFcShort(0x0),	/* 0 */
FC_SHORT */	0x6,	/*	[pointer_deref] */		0x12, 0x10, /* FC_UP	/* 936 */ NdrFcShort(0x0),	/* Offset= 0 (936) */
/* 850 */ 0x38,	/* FC_ALIGNM4 */	/*	/* 892 */ NdrFcShort(0xffffffffdbc),	/* Offset= -580 (312) */	0x12, 0x10, /* FC_UP	/* 938 */ 0x8,	/* FC_LONG */
FC_LONG */	0x8,	/*	/* 894 */	/* FC_UP	0x12, 0x10, /* FC_UP	FC_LONG */	/*
/* 852 */ 0x8,	/* FC_LONG */	/*	[pointer_deref] */		0x12, 0x10, /* FC_UP	/* 940 */ 0x6,	/* FC_SHORT */
FC_EMBEDDED_COMPLEX */	0x4c,	/*	/* 896 */ NdrFcShort(0xffffffffdca),	/* Offset= -566 (330) */	0x12, 0x10, /* FC_UP	/* 942 */ 0x6,	/* FC_SHORT */
/* 854 */ 0x4,	/* 4 */		/* 898 */	/* FC_UP	0x12, 0x10, /* FC_UP	FC_SHORT */	/*
Offset= -499 (356) */	NdrFcShort(0xffffffffe0d),	/*	[pointer_deref] */		0x12, 0x10, /* FC_UP	/* 944 */ 0x4c,	/* FC_EMBEDDED_COMPLEX
FC_END */	0x5b,	/*	/* 900 */ NdrFcShort(0xffffffffdd8),	/* Offset= -552 (348) */	0x12, 0x10, /* FC_UP	*/	/*
/* 858 */			/* 902 */	/* FC_UP	0x12, 0x10, /* FC_UP	0x0,	/* 0
/* 860 */ NdrFcShort(0xfffffffff02),	0x12, 0x0, /* FC_UP */		[pointer_deref] */		0x12, 0x10, /* FC_UP	/* 946 */ NdrFcShort(0xffffffffc54),	/* Offset= -940 (6) */
/* 862 */	/* Offset= -254 (606) */		/* 904 */ NdrFcShort(0x2),	/* Offset= 2 (906) */	0x12, 0x0, /* FC_UP */	/* 948 */ 0x5c,	/* FC_PAD */
[simple_pointer] */	0x12, 0x8, /* FC_UP		/* 906 */	/* FC_UP	0x12, 0x0, /* FC_UP */	0x5b,	/*
/* 864 */ 0x1,	/* FC_BYTE */		/* 908 */ NdrFcShort(0x16),	/* Offset= 22 (930) */	0x15,	FC_END */	/*
FC_PAD */	0x5c,	/*	/* 910 */	/* FC_UP */	0x7,	/* 950 */ 0xb4,	/* FC_USER_MARSHAL */
/* 866 */	0x12, 0x8, /* FC_UP		FC_STRUCT */		/* 912 */ NdrFcShort(0x10),	131 */	/*
[simple_pointer] */			*/		/* 914 */ 0x6,	/* 952 */ NdrFcShort(0x0),	/* 0 */
			/* 912 */	/* FC_SHORT */	/* 914 */	/* 954 */ NdrFcShort(0x18),	/* 24 */
			/* 914 */			/* 956 */ NdrFcShort(0x0),	/* 0 */
						/* 958 */ NdrFcShort(0xffffffffc44),	/* Offset= -956 (2) */
						/* 960 */	

```

                                0x11, 0x4, /* FC_RP
[allocated_on_stack] */
/* 962 */ NdrFcShort( 0x6 ), /* Offset= 6 (968) */
/* 964 */
                                0x13, 0x0, /* FC_OP */
/* 966 */ NdrFcShort( 0xfffffdc ), /* Offset= -36 (930) */
/* 968 */ 0xb4, /* FC_USER_MARSHAL */
                                0x83, /*
131 */
/* 970 */ NdrFcShort( 0x0 ), /* 0 */
/* 972 */ NdrFcShort( 0x18 ), /* 24 */
/* 974 */ NdrFcShort( 0x0 ), /* 0 */
/* 976 */ NdrFcShort( 0xfffffff4 ), /* Offset= -12 (964) */

                                0x0

    }
};

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

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

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

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

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex
)
{
    if(!_tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }

    return 0;
}

```

```

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

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

```

tpcc_com_ps\src\dlldata.c

```

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

    DO NOT ALTER THIS FILE

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

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

*****/

#include <rpcproxy.h>

#ifdef __cplusplus
extern "C" {
#endif

EXTERN_PROXY_FILE( tpcc_com_ps )

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

```

```

DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )

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

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

```

Appendix B Database Load

Setup.cmd

@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=8
set WAREHOUSES=26000
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',32" -b
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"reconfigure with override" -b
if ERRORLEVEL 1 goto :ERROR_EXIT
```

```
rem goto :BYPASS
```

```
echo Dropping and Creating Tables %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
```

```
rem :BYPASS
echo Starting bulk inserts of item customer warehouse orders
rem echo Starting bulk inserts of orders
```

```
for %i in (warehouse customer orders) do call :BULKINSERTi %i
rem for %i in (orders) do call :BULKINSERTi %i
if ERRORLEVEL 1 goto :ERROR_EXIT
```

```
goto :CLEANUP
```

```
:BULKINSERTi
echo Starting bulk inserts for %1
for /l %j in (1,1,%THREADS%) do start cmd /C
%HOMEPATH%\SetupOldLoader BULKINSERTn %1 %j
%EXECUTABLEPATH%\Semaphore -wait %THREADS%
```

```
echo Starting create indexes for %1
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\tpcldr -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 -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: TPC.H
// 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 <sys\types.h>
#include <math.h>

// ODBC headers
```

```
#include <sql.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 ".\\log";
#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

    BOOL table_item; // set if loading ITEM

    BOOL table_specific;
    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;
        long
        retry_limit;
        long
        sleep_milli;
} TPCCLDR_ARGS;

// String length constants
#define SERVER_NAME_LEN      20
#define DATABASE_NAME_LEN    20
#define USER_NAME_LEN        20
#define PASSWORD_LEN         20
#define TABLE_NAME_LEN      20
#define I_DATA_LEN           50
#define I_NAME_LEN           24
#define BRAND_LEN             1
#define LAST_NAME_LEN         16
#define W_NAME_LEN            10
#define ADDRESS_LEN           20
#define STATE_LEN             2
#define ZIP_LEN                9
#define S_DIST_LEN            24
#define S_DATA_LEN            50
#define D_NAME_LEN            10

```

```

#define MIDDLE_NAME_LEN      16
#define PHONE_LEN            16
#define CREDIT_LEN           2
#define C_DATA_LEN           500
#define H_DATA_LEN           24
#define DIST_INFO_LEN        24
#define MAX_OL_NEW_ORDER_ITEMS 15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN           25
#define OL_DIST_INFO_LEN      24

#define C_SINCE_LEN          23
#define H_DATE_LEN           23
#define OL_DELIVERY_D_LEN    23
#define O_ENTRY_D_LEN        23

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

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

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

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

tpccldr.c

// File: TPCCLDR.C

```

```

// Microsoft TPC-C Kit
// Ver. 4.42 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

```



```

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

        // process command line arguments

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

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

        if (aptr->build_index == 0)
            printf("Data load only - no index creation.\n");
        else
        {
            printf("Data load and index creation.\n");
            if (aptr->index_order == 0)
                printf("Clustered indexes will be
                created after bulk load.\n");
            else
                printf("Clustered indexes will be
                created before bulk load.\n");
        }

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

        // open connections to SQL Server

```

```

        OpenConnections();

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

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

        // start loading data

        sprintf(buffer, "TPC-C load started for %ld warehouses. 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)
            {

```

```

orders\n");

        fprintf(floader, "Starting loader threads for:

        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 != SUCCEEDED)
                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 != SUCCEEDED)
                        HandleErrorDBC(i_hdbc1);

```

```

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

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

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

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

        rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0, I_DATA_LEN, NULL,
0, 0, 5);
        if (rc != SUCCEEDED)
                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 != SUCCEEDED)
                        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, "warehouse.err");
        // sprintf(err_log_path, "%swarehouse%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("idxdiscl");

    InitString(d_name, D_NAME_LEN+1);
    InitAddress(d_street_1, d_street_2, d_city, d_state, d_zip);
    sprintf(name, "%s..%s", aptr->database, "district");

```

```

//      strcpy(err_log_path, aptr->err_log_path);
    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;
    char    name[20];
    long    time_start;
    RETCODE rc;
    DBINT   rcint;
    char    bcphint[128];
    char    err_log_path[256];

    // Seed with unique number
    seed(3);

```

```

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

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

//rc = bcp_init(w_hdbc1, name, NULL, "logs\\stock.err",
DB_IN);
//
// strcpy(err_log_path,aptr->log_path);
// strcat(err_log_path,"stock.err");
// 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(bcphint, "tablock, ROWS_PER_BATCH =
%u", (aptr->num_warehouses * 100000));
    rc = bcp_control(w_hdbc1, BCPHINTS, (void*)
bcphint);

    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
}

rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

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("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");
}

// Initialize bulk copy
sprintf(name, "%s..%s", aptr->database, "customer");

DB_IN);
//
// rc = bcp_init(c_hdbc1, name, NULL, "logs\\customer.err",
//
// 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(bcpint, "tablock");
        rc = bcp_control(c_hdbc2, BCPHINTS, (void*) bcpint);
        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,

                                0,

(LPTHREAD_START_ROUTINE) LoadCustomerTable,

&customer_time_start,

                                0,

                                &dwThreadID[0]);

```

```

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

        // Start History table thread

        printf("...Loading history table for:
d_id = %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("idxcust");
        // 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("idxhisc");
    }

    // 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",
                                aptr-
>server,
                                aptr-
>user,
                                aptr-
>password,
                                aptr-
>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,"");
        strcpy(customer_buf[i].c_street_1,"");
        strcpy(customer_buf[i].c_street_2,"");
        strcpy(customer_buf[i].c_city,"");
        strcpy(customer_buf[i].c_state,"");
        strcpy(customer_buf[i].c_zip,"");
        strcpy(customer_buf[i].c_phone,"");
        strcpy(customer_buf[i].c_credit,"");

        customer_buf[i].c_credit_lim = 0;
        customer_buf[i].c_discount = (float) 0;
    }
}

```

aptr-

```

problem.

// 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
            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 =
%ld\n",
        d_id, w_id);

for (i=0;i<customers_per_district;i++)
{
    customer_buf[i].c_d_id = d_id;
    customer_buf[i].c_w_id = w_id;
    customer_buf[i].h_amount = 10.0;

    customer_buf[i].c_ytd_payment = 10.0;

    customer_buf[i].c_payment_cnt = 1;
    customer_buf[i].c_delivery_cnt = 0;

    // Generate CUSTOMER and HISTORY data

    customer_buf[i].c_id = c[i].c_id;

    strcpy(customer_buf[i].c_first, c[i].c_first);
    strcpy(customer_buf[i].c_last, c[i].c_last);

    customer_buf[i].c_middle[0] = 'O';
    customer_buf[i].c_middle[1] = 'E';

    MakeAddress(customer_buf[i].c_street_1,
customer_buf[i].c_street_2,
customer_buf[i].c_city,
customer_buf[i].c_state,
customer_buf[i].c_zip);

    MakeNumberString(16, 16, PHONE_LEN,
customer_buf[i].c_phone);

    if (RandomNumber(1L, 100L) > 10)
        customer_buf[i].c_credit[0] = 'G';
    else
        customer_buf[i].c_credit[0] = 'B';
    customer_buf[i].c_credit[1] = 'C';

    customer_buf[i].c_credit_lim = 50000.0;

    customer_buf[i].c_discount = ((float)
RandomNumber(0L, 5000L)) / 10000.0;
}

```

```

        // fix to avoid ODBC float to numeric conversion
problem.
        // customer_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 != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 2);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 3);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0, FIRST_NAME_LEN, NULL, 0,
0, 4);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0, MIDDLE_NAME_LEN, NULL,
0, 0, 5);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN, NULL, 0, 0,
6);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN, NULL, 0,
0, 7);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN, NULL, 0, 0,
8);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL, 0, 0, 9);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0, 0, 10);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0, 11);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

```

```

        rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL, 0, 0,
12);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0, C_SINCE_LEN,
NULL, 0, SQLCHARACTER, 13);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL, 0, 0,
14);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 15);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 16);
        if (rc != SUCCEEDED)
            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 != SUCCEEDED)
        //     HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0,
SQLCHARACTER, 17);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 18);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 19);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);

        rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 20);
        if (rc != SUCCEEDED)
            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
            // c_balance = customer_buf[i].c_balance;
            strcpy(c_balance, customer_buf[i].c_balance);

            c_ytd_payment =
customer_buf[i].c_ytd_payment;
            c_payment_cnt =
customer_buf[i].c_payment_cnt;
            c_delivery_cnt = customer_buf[i].c_delivery_cnt;

            strcpy(c_data, customer_buf[i].c_data);

            // Send data to server
            rc = bcp_sendrow(c_hdbc1);
            if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

            customer_rows_loaded++;
            CheckForCommit(c_hdbc1, c_hstmt1,
customer_rows_loaded, "customer", &customer_time_start->time_start);
        }
    }
}

```

```

//=====
//
// Function : LoadHistoryTable
//
//=====

void LoadHistoryTable(LOADER_TIME_STRUCT *history_time_start)
{
    long i;
    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 w_id;
    short d_id;
    DWORD dwThreadId[MAX_ORDER_THREADS];
    HANDLE hThread[MAX_ORDER_THREADS];
    char name[20];

```

```

RETCODE
rc;
char
bcphint[128];
char
err_log_path_ord[256];
char
err_log_path_nord[256];
char
err_log_path_ordl[256];

// seed with unique number
seed(6);

printf("Loading orders...\n");

// if build index before load...
if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    BuildIndex("idxordcl");
    BuildIndex("idxnordcl");
    BuildIndex("idxordlcl");
}

// 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)
        {
            orders_buf[o_id].o_carrier_id =

(short)RandomNumber(1L, 10L);

            orders_buf[o_id].o_all_local = 1;
        }
        else
        {
            orders_buf[o_id].o_carrier_id = 0;
            orders_buf[o_id].o_all_local = 1;
        }

        for (ol=0; ol<orders_buf[o_id].o_ol_cnt; ol++)
    }

```

```

    {
        orders_buf[o_id].o_ol[ol].ol = ol+1;
        orders_buf[o_id].o_ol[ol].ol_i_id =
RandomNumber(1L, max_items);

        orders_buf[o_id].o_ol[ol].ol_supply_w_id = w_id;
        orders_buf[o_id].o_ol[ol].ol_quantity
= 5;
        MakeAlphaString(24, 24,
OL_DIST_INFO_LEN, &orders_buf[o_id].o_ol[ol].ol_dist_info);

        // Generate ORDER-LINE data
        if (o_id < first_new_order)
        {

            orders_buf[o_id].o_ol[ol].ol_amount = 0;
            // Added to insure
ol_delivery_d set properly during load

            FormatDate(&orders_buf[o_id].o_ol[ol].ol_delivery_d);

        }
        else
        {

            orders_buf[o_id].o_ol[ol].ol_amount =
RandomNumber(1,999999)/100.0;
            // Added to insure
ol_delivery_d set properly during load

            // odbc datetime

            format

            strcpy(orders_buf[o_id].o_ol[ol].ol_delivery_d,"1899-12-31
00:00:00.000");

        }
    }
}

//=====
//
// Function : LoadOrdersTable
//

```

```

//=====
//=====
void LoadOrdersTable(LOADER_TIME_STRUCT *orders_time_start)
{
    int i;
    long o_id;
    short o_d_id;
    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("idxordcl");

        // build non-clustered index

```

```

        if (aptr->build_index == 1)
            BuildIndex("idxordnc");
    }

//=====
//
// Function : LoadNewOrderTable
//
//=====

void LoadNewOrderTable(LOADER_TIME_STRUCT *new_order_time_start)
{
    long          i;
    long o_id;
    short 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 != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 2);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 3);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);

    for (i = first_new_order; i < last_new_order; i++)
    {
        o_id = orders_buff[i].o_id;
        o_d_id = orders_buff[i].o_d_id;
        o_w_id = orders_buff[i].o_w_id;

        rc = bcp_sendrow(o_hdbc2);
        if (rc != SUCCEEDED)

            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("idxnodd!");
    }

//=====
//
// Function : LoadOrderLineTable
//
//=====

void LoadOrderLineTable(LOADER_TIME_STRUCT *order_line_time_start)
{
    long          i;
    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_dist_info[OL_DELIVERY_D_LEN+1];
DBINT rcint;

// bind ORDER-LINE data
rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 2);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 3);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 4);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 5);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 6);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0,
OL_DELIVERY_D_LEN, NULL, 0, SQLCHARACTER, 7);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 8);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 9);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0, DIST_INFO_LEN, NULL,
0, 0, 10);
if (rc != SUCCEEDED)

```

```

        HandleErrorDBC(o_hdbc3);

        for (i = 0; i < orders_per_district; i++)
        {
            o_id = orders_buff[i].o_id;
            o_d_id = orders_buff[i].o_d_id;
            o_w_id = orders_buff[i].o_w_id;

            for (j=0; j < orders_buff[i].o_ol_cnt; j++)
            {
                ol = orders_buff[i].o_ol[j].ol;
                ol_i_id =

                orders_buff[i].o_ol[j].ol_i_id;
                ol_supply_w_id =
                orders_buff[i].o_ol[j].ol_supply_w_id;
                ol_quantity =
                orders_buff[i].o_ol[j].ol_quantity;
                ol_amount =
                orders_buff[i].o_ol[j].ol_amount;

                strcpy(ol_delivery_d,orders_buff[i].o_ol[j].ol_delivery_d);

                strcpy(ol_dist_info,orders_buff[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);
        }
    }
}

```

```

SQLDisconnect(o_hdbc3);
SQLFreeConnect(o_hdbc3);

// if build index after load...
if ((aptr->build_index == 1) && (aptr-
>index_order == 0))

    BuildIndex("idxodid");

}

}

//=====
//
// 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(c_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_hdbc2);
                    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],

```

```

        {
            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,
        Msg, sizeof(Msg) ,
    &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
    char          szDriverString[300];
    char          szDriverStringOut[1024];
    char          TablesBitMap[9] =
{"000000000"};
    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");
        printf("\nCheck LOGS\\ directory for database
creation errors.\n");

        // cleanup database connections and handles
        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
        SQLDisconnect(v_hdbc);
        SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

        // since there is not a database, exit back to
        SETUP.CMD
    }

```

```

        exit(1);
    }

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

    if ( SQLBindCol(v_hstmt, 1, SQL_C_ULONG, &TabCount, 0,
    &TabCountInd) != SQL_SUCCESS )
        HandleErrorSTMT(v_hstmt);

    // count the number of user tables from sysobjects
    rc = SQLExecDirect(v_hstmt, "select count(*) from sysobjects
    where xtype = 'U'", SQL_NTS);
    if ((rc != SQL_SUCCESS) && (rc !=
    SQL_SUCCESS_WITH_INFO))
        HandleErrorSTMT(v_hstmt);

    if ( SQLFetch(v_hstmt) != SQL_SUCCESS )
        HandleErrorSTMT(v_hstmt);

    // if the number of tables is less than 9, select all the user
    tables in TPCC
    if (TabCount != 9)
    {
        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);

        SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc ,
        &v_hstmt);

        if ( SQLBindCol(v_hstmt, 1, SQL_C_CHAR,
        &TabName, sizeof(TabName), &TabNameInd) != SQL_SUCCESS )
            HandleErrorSTMT(v_hstmt);

        // select the list of user tables into a result set
        rc = SQLExecDirect(v_hstmt, "select * from
        sysobjects where xtype = 'U'", SQL_NTS);
        if ((rc != SQL_SUCCESS) && (rc !=
        SQL_SUCCESS_WITH_INFO))
            HandleErrorSTMT(v_hstmt);

        // go through the result set and set the bitmap for
        each found table
        // set the bitmap to '1' if the table name is found
        while ((rc = SQLFetch(v_hstmt)) !=
        SQL_NO_DATA)
        {
            switch( TabName[0] )
            {
                case 'w':

```

```

                break;
            case 'd':
                TablesBitMap[0] = '1';
                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
= 1;                                }
                                break;
                                case 7:
                                if (TablesBitMap[i] ==
'0')
                                {
                                printf("The Item table is missing or damaged.\n");
                                ExitFlag
= 1;                                }
                                break;
                                case 8:
                                if (TablesBitMap[i] ==
'0')
                                {
                                printf("The Stock table is missing or damaged.\n");
                                ExitFlag
= 1;                                }
                                break;
                                }
                                // if one or more tables are missing, display
message and exit the loader
                                if (ExitFlag = 1)
                                {
                                printf("\nExiting TPC-C Loader!\n");
                                printf("\nCheck LOGS\ directory for
database\n");
                                printf("or table creation errors.\n");

                                // cleanup database connections and
handles
                                SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
                                SQLDisconnect(v_hdbc);
                                SQLFreeHandle(SQL_HANDLE_DBC,
v_hdbc);

                                exit(1);
                                }

                                // cleanup database connections and handles
                                SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
                                SQLDisconnect(v_hdbc);

```

```

SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);
return;
}

```

time.c

```

//      File:              TIME.C
//      Ver. 4.42          Microsoft TPC-C Kit
//      Copyright Microsoft,
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          Microsoft TPC-C Kit
//      Copyright Microsoft,
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("[%d]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("[%d]DBG: LastName: num = [%d] ==> [%d][%d][%d]\n",
              (int) GetCurrentThreadId(), num,
num/100, (num/10)%10, num%10);
    printf("[%d]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("[%d]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("[%d]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);
    }
}

```

```

        strcpy(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:
*
* 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
//Orginal 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, TPCCLDR_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_customer = 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':
                pargs->server =
                ptr+2;
                break;

            case 'U':
                pargs->user = ptr+2;
                break;

            case 'b':

```

<pre> atol(ptr+2); case 'W': >num_warehouses = atol(ptr+2); case 's': >starting_warehouse = atol(ptr+2); case 't': { >tables_all = FALSE; (strcmp(ptr+2,"item") == 0) pargs->table_item = TRUE; (strcmp(ptr+2,"warehouse") == 0) pargs->table_warehouse = TRUE; (strcmp(ptr+2,"customer") == 0) pargs->table_customer = TRUE; (strcmp(ptr+2,"orders") == 0) pargs->table_orders = TRUE; printf("\nUnrecognized command"); GetArgsLoaderUsage(); exit(1); } case 'f': = ptr+2; case 'L': </pre>	<pre> pargs->batch = break; pargs- break; pargs- break; { pargs- if else if else if else if else { exit(1); } break; } pargs->loader_res_file break; case 'L': </pre>	<pre> ptr+2; atol(ptr+2); atol(ptr+2); atol(ptr+2); atol(ptr+2); atol(ptr+2); >index_script_path = ptr+2; atol(ptr+2); atol(ptr+2); } /* check for required args */ if (pargs->num_warehouses == UNDEF) { printf("Number of Warehouses is required\n"); exit(-2); } return; </pre>	<pre> pargs->log_path = break; case 'p': pargs->pack_size = break; case 'i': pargs->build_index = break; case 'o': pargs->index_order = break; case 'c': pargs->scale_down = break; pargs- break; case 'R': pargs->retry_limit = break; case 'r': pargs->sleep_milli = break; default: GetArgsLoaderUsage(); exit(-1); break; } } </pre>	<pre> } //===== // // Function name: GetArgsLoaderUsage // //===== void GetArgsLoaderUsage() { #ifdef DEBUG printf("[%ld]DBG: Entering GetArgsLoaderUsage()\n", (int) GetCurrentThreadId()); #endif printf("TPCCLDR:\n\n"); printf("Parameter Default\n"); printf("-----\n"); printf("-W Number of Warehouses to Load Required \n"); printf("-S Server %s\n", SERVER); printf("-U Username %s\n", USER); printf("-P Password %s\n", PASSWORD); printf("-D Database %s\n", DATABASE); printf("-b Batch Size %ld\n", (long) BATCH); printf("-p TDS packet size %ld\n", (long) DEFLDPACKSIZE); printf("-f Loader Results Output Filename %s\n", LOADER_RES_FILE); printf("-s Starting Warehouse %ld\n", (long) DEF_STARTING_WAREHOUSE); printf("-i Build Option (data = 0, data and index = 1) %ld\n", (long) BUILD_INDEX); printf("-o Cluster Index Build Order (before = 1, after = 0) %ld\n", (long) INDEX_ORDER); printf("-R Retry limit. # of times to retry open %ld\n", (long) RETRYLIMIT); printf("-r Retry sleep time in milliseconds. %ld\n", (long) SLEEPMILLI); 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"); </pre>
--	--	--	--	--

```

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. 4.20
--         Copyright Microsoft, 1999
-- Purpose: Sets database options for data load

use master
go

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

use tpcc
go

alter database tpcc set recovery simple
go

checkpoint
go

```

DBOPT2 . SQL

```

-- File:   DBOPT2_SHILOH.SQL
--         Microsoft TPC-C Benchmark Kit Ver. 4.10a
--         Copyright Microsoft, 1999
-- Purpose: Resets database options after data load (For Shiloh)

use master
go

sp_dboption tpcc,'select ',false
go

sp_dboption tpcc,'trunc. ',false
go

use tpcc
go

checkpoint
go

alter database tpcc set recovery full
go

sp_configure allow,1
go

reconfigure with override
go

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

sp_indexoption 'customer', 'DisallowPageLocks', TRUE
go
sp_indexoption 'district', 'DisallowPageLocks', TRUE
go
sp_indexoption 'warehouse', 'DisallowPageLocks', TRUE
go
sp_indexoption 'stock', 'DisallowPageLocks', TRUE
go
sp_indexoption 'order_line', 'DisallowRowLocks', TRUE
go
sp_indexoption 'orders', 'DisallowRowLocks', TRUE
go

```

```

sp_indexoption 'new_order', 'DisallowRowLocks', TRUE
sp_indexoption 'item', 'DisallowRowLocks', TRUE
go

sp_indexoption 'item', 'DisallowPageLocks', TRUE
go

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

select name,lockflags
from sysindexes
where object_id("warehouse") = id or
       object_id("district") = id or
       object_id("customer") = id or
       object_id("stock") = id or
       object_id("orders") = id or
       object_id("order_line") = id or
       object_id("history") = id or
       object_id("new_order") = id or
       object_id("item") = id
order by lockflags asc
go

sp_configure allow,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

```
--
-- HPTPC 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 NOT NULL,
    w_name char(10) NOT NULL,
    w_street_1 char(20) NOT NULL,
    w_street_2 char(20) NOT NULL,
    w_city char(20) NOT NULL,
    w_state char(2) NOT NULL,
    w_zip char(9) NOT NULL,
    w_tax numeric(4,4) NOT NULL,
    w_ytd numeric(12,2) NOT NULL
) on load_fg
go
```

```
create table district
(
    d_id tinyint NOT NULL,
    d_w_id int NOT NULL,
    d_name char(10) NOT NULL,
    d_street_1 char(20) NOT NULL,
    d_street_2 char(20) NOT NULL,
    d_city char(20) NOT NULL,
    d_state char(2) NOT NULL,
    d_zip char(9) NOT NULL,
    d_tax numeric(4,4) NOT NULL,
    d_ytd numeric(12,2) NOT NULL,
    d_next_o_id int NOT NULL
) on load_fg
go
```

```
create table customer
(
    c_id int NOT NULL,
    c_d_id tinyint NOT NULL,
    c_w_id int NOT NULL
)
```

```
    c_first char(16) NOT NULL,
    c_middle char(2) NOT NULL,
    c_last char(16) NOT NULL,
    c_street_1 char(20) NOT NULL,
    c_street_2 char(20) NOT NULL,
    c_city char(20) NOT NULL,
    c_state char(2) NOT NULL,
    c_zip char(9) NOT NULL,
    c_phone char(16) NOT NULL,
    c_since datetime NOT NULL,
    c_credit char(2) NOT NULL,
    c_credit_lim numeric(12,2) NOT NULL,
    c_discount numeric(4,4) NOT NULL,
    c_balance numeric(12,2) NOT NULL,
    c_ytd_payment numeric(12,2) NOT NULL,
    c_payment_cnt smallint NOT NULL,
    c_delivery_cnt smallint NOT NULL,
    c_data char(500) NOT NULL
) on load_fg
go

create table history
(
    h_c_id int NOT NULL,
    h_c_d_id tinyint NOT NULL,
    h_c_w_id int NOT NULL,
    h_d_id tinyint NOT NULL,
    h_w_id int NOT NULL,
    h_date datetime NOT NULL,
    h_amount numeric(6,2) NOT NULL,
    h_data char(24) NOT NULL
) on load_fg
go
```

```

create table new_order
(
    no_o_id          int NOT
NULL,
    no_d_id          tinyint
NOT NULL,
    no_w_id          int NOT
NULL
) on load_fg
go

create table orders
(
    o_id            int NOT
NULL,
    o_d_id          tinyint
NOT NULL,
    o_w_id          int NOT
NULL,
    o_c_id          int NOT
NULL,
    o_entry_d        datetime NOT NULL,
    o_carrier_id     tinyint NOT NULL,
    o_ol_cnt         tinyint NOT NULL,
    o_all_local      tinyint NOT NULL
) on load_fg
go

create table order_line
(
    ol_o_id          int NOT
NULL,
    ol_d_id          tinyint
NOT NULL,
    ol_w_id          int NOT
NULL,
    ol_number        tinyint NOT NULL,
    ol_i_id          int NOT
NULL,
    ol_supply_w_id   int NOT
NULL,
    ol_delivery_d     datetime NOT NULL,
    ol_quantity       smallint NOT NULL,
    ol_amount         numeric(6,2) NOT
NULL,
    ol_dist_info      char(24) NOT NULL
) on load_fg
go

create table item
(

```

```

    i_id            int NOT
NULL,
    i_im_id          int NOT
NULL,
    i_name           char(24)
NOT NULL,
    i_price          numeric(5,2) NOT NULL,
    i_data           char(50)
NOT NULL
) on load_fg
go

create table stock
(
    s_i_id          int NOT
NULL,
    s_w_id          int NOT
NULL,
    s_quantity       smallint NOT NULL,
    s_dist_01        char(24) NOT NULL,
    s_dist_02        char(24) NOT NULL,
    s_dist_03        char(24) NOT NULL,
    s_dist_04        char(24) NOT NULL,
    s_dist_05        char(24) NOT NULL,
    s_dist_06        char(24) NOT NULL,
    s_dist_07        char(24) NOT NULL,
    s_dist_08        char(24) NOT NULL,
    s_dist_09        char(24) NOT NULL,
    s_dist_10        char(24) NOT NULL,
    s_ytd            int NOT
NULL,
    s_order_cnt       smallint NOT NULL,
    s_remote_cnt      smallint
NOT NULL,
    s_data           char(50)
NOT NULL
) on 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
tableoption 'stock','table lock on bulk load','on'
go

```

newcreateCustomerindexes.sql

```

--
-- HP TPCC Setup
-- Create Indexes for Customer
-- and move the table from LoadFg to CustomerStockFg
--
use tpcc
go

create unique clustered index customer_c1 on customer(c_w_id, c_id,
c_d_id)
with SORT_IN_TEMPDB, fillfactor=100 on cs_fg
go

--create unique nonclustered index customer_nc1 on customer(c_w_id,
c_d_id, c_last, c_first, c_id)
--with SORT_IN_TEMPDB, fillfactor=100
--go

--create unique clustered index history_c1 on history(h_c_w_id, h_date,
h_c_d_id, h_c_id, h_amount)
-- with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
--go

create clustered index history_c1 on history(h_c_w_id, h_date, h_c_d_id,
h_c_id, h_amount)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go

--

```

newcreateItemindexes.sql

```

-- HPTPCC Setup Kit
-- Create Item Indexes
-- and move the table from LoadFg to MiscFg
--
use tpcc
go

```

```
create unique clustered index item_c1 on item(i_id)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go
```

newcreateOrdersindexes.sql

```
--
-- HPTPCC Setup Kit
-- Create ORDERS Indexes
-- and move the table from LoadFg to MiscFg
--
use tpcc
go

create unique clustered index orders_c1 on orders(o_w_id, o_d_id, o_id)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go

create index orders_nc1 on orders(o_w_id, o_d_id, o_c_id, o_id)
with SORT_IN_TEMPDB, fillfactor=100
go

create unique clustered index new_order_c1 on new_order(no_w_id,
no_d_id, no_o_id)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go

create unique clustered index order_line_c1 on order_line(ol_w_id, ol_d_id,
ol_o_id, ol_number)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go
```

newcreateStockindexes.sql

```
--
-- HPTPCC Setup Kit
-- Create STOCK Indexes
-- and move the table from LoadFg to MiscFg
--
use tpcc
go
```

```
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go
```

```
create unique clustered index district_c1 on district(d_w_id, d_id)
with fillfactor=100, SORT_IN_TEMPDB on misc_fg
go
```

```
create unique clustered index stock_c1 on stock(s_i_id,s_w_id)
with SORT_IN_TEMPDB, fillfactor=100 on cs_fg
go
```

newcreateWarehouseindexes.sql

```
--
-- HPTPCC Setup Kit
-- Create WAREHOUSE Indexes
-- and move the table from LoadFg to MiscFg
--
```

```
use tpcc
go
```

```
create unique clustered index warehouse_c1 on warehouse(w_id)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go
```

```
create unique clustered index district_c1 on district(d_w_id, d_id)
with fillfactor=100, SORT_IN_TEMPDB on misc_fg
go
```

```
create unique clustered index stock_c1 on stock(s_i_id,s_w_id)
with SORT_IN_TEMPDB, fillfactor=100 on cs_fg
go
```

B.3 Stored Procedures

createnewordproc.sql

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

```
== 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
```

```

int,                                     @w_id
tinyint,                                @d_id
int,                                     @c_id

        @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)

@i_id14          begin
                  select @li_no = @li_no + 1

@i_id15          -----
                  -- 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 - @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 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

```



```

        @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 = 'sp1tpcc_neworder' )
        drop procedure sp1tpcc_neworder

go

create proc sp1tpcc_neworder

        @w_id
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 = 'sp2tpcc_neworder' )
    drop procedure sp2tpcc_neworder
go
create proc sp2tpcc_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,

```

```

                                @w_id)
                                int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
                                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

```

155

```

        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 = '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

        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

@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15

               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

               -----
               -- 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

when
10 then s_dist_10

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 = '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,
    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),
           @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
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 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,
    @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 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
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) )
            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 = 'sp6tpcc_neworder' )
drop procedure sp6tpcc_neworder
go
create proc sp6tpcc_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

              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

              @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,
                          @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

```

```

then s_dist_08
when 8
then s_dist_09
when 9
10 then s_dist_10
when
end
= @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 * @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
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 = 'sp7tpcc_neworder' )
drop procedure sp7tpcc_neworder
go
create proc sp7tpcc_neworder
int,
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,

```

```

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
begin transaction n

```

```

@i_id5
@i_id6
@i_id7
@i_id8
@i_id9
@i_id10
@i_id11
@i_id12
@i_id13
@i_id14
@i_id15

-- 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

```

```

@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8
@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15

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
@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_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,
@s_price,
@s_price * @li_qty

end
else
begin
-----
-- no item (or stock) found - triggers rollback condition
-----

```

```

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
```

```
            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 = '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_d_id       = @d_id and

        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 = '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
            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

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 = '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
    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
    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
    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 = '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_middle = c_middle,
        @o_id     = o_id,
        @o_entry_d = o_entry_d,
        @o_carrier_id = o_carrier_id,
        @cnt
    char(16),
    int,
    datetime,
    smallint,
    smallint

begin tran o

if (@c_id = 0)
begin
-----
-- get customer id and info using last name
-----
    select
        @cnt = (count(*)+1)/2
    from
    where
    customer WITH (repeatableread)
    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
        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
    where
    customer WITH (repeatableread)
    c_id = @c_id

and

```

```

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,

```

```

    @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_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,
                  c_w_id  = @w_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 = '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
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
```

```
order       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 = '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        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_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 = '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),
        @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
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
-----

```

```

begin
if @cnt = 0)
    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

```

createdeliveryproc.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 int,
                        @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
as

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,
                                @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 =
                                @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
                                = 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 = 'sp0tpcc_delivery' )
                                drop procedure sp0tpcc_delivery
                                go

```

```

create proc sp0tpcc_delivery @w_id int,
                                @o_carrier_id int,
                                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

= @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

where

= @total + ol_amount

= @w_id and

= @d_id and

= @o_id

-----
-- accumulate lineitem amounts for this order into customer
-----
update customer
set c_balance =

c_delivery_cnt

where

= c_delivery_cnt + 1

= @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,
@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),
@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

= @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

= @total + ol_amount

```

```

        = @w_id and      where      ol_w_id
        = @d_id and      ol_d_id
        = @o_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
        where      c_w_id
        = @w_id and
        = @d_id and
        = @o_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
                        @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
-----

```

```

and
and
and
        where      new_order      = @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
        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
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,

```

```

        @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

                                = o_c_id

                                where

                                = @w_id and

                                = @d_id and

                                = @o_id

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

```

```

-----update-----
set          order_line
              ol_delivery_d =

getdate(),

              @total

              = @total + ol_amount

              where

              = @w_id and

              = @d_id and

              = @o_id

              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

              where

              = @w_id and

              = @d_id and

              = @c_id

              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 = '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,

updlock)

        where   no_w_id   = @w_id and
                no_d_id   = @d_id
        order   by no_o_id asc

```

```

        if (@@trancount <> 0)

        -----
        -- 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

                = @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

```

```

                c_d_id
                c_id

        = @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 = '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

            and

            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

            where o_w_id

            = @w_id and

```

```

                = @d_id and
                = @o_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

            = 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 = '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

```

```

uplock)
    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
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
where ol_w_id
ol_d_id
ol_o_id

-----
-- accumulate lineitem amounts for this order into customer
-----

```

```

set date c_balance =
c_balance + @total,
c_delivery_cnt =
c_delivery_cnt + 1
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 = '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 (@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
= @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

```

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_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_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,

```

```

where       @c_id_local = 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(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,

```

```

where       @d_id_local = d_id = d_name,
            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 = @w_id

where       w_id = @w_id

-----
-- create history record
-----
insert into history values ( @c_id_local,
@c_d_id,
@d_id_local,
@w_id_local,
@datetime,
@c_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 = '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_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),
@c_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,
@c_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_city,
       @w_state,
       @w_zip,
       @d_street_1,
       @d_street_2,
       @d_city,
       @d_state,
       @d_zip,
       @d_name,
       @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 = '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_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,
            @c_data        = c_data,

            @c_id_local   = c_id

```

```

where       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,

```

```

where       @w_id_local = d_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 = @w_id

where       w_id

-----
-- create history record
-----
insert into history values ( @c_id_local,
@c_d_id,
@c_w_id,

@c_id_local,

@c_w_id_local,

@datetime,

@c_h_amount,

@c_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 = 'sp2tpcc_payment')
drop procedure sp2tpcc_payment
go

create proc sp2tpcc_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_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,
@c_data = c_data,
@c_id_local = c_id
where c_id = @c_id and
c_w_id = @c_w_id and
c_d_id = @c_d_id

-----
-- if customer has bad credit get some more info
-----
if (@c_credit = 'BC')
begin
-----
-- compute new info
-----
select @c_data =
convert(char(5),@c_id) +
convert(char(4),@c_d_id) +
convert(char(5),@c_w_id) +
convert(char(4),@d_id) +
convert(char(5),@w_id) +
convert(char(19),@h_amount) +
substring(@data, 1,
458)
-----

```

```

-- update customer info
-----
update      customer
set         c_data      = @c_data

where       c_id        = @c_id and
            c_w_id      = @c_w_id and
            c_d_id      = @c_d_id

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

-----
-- get district data and update year-to-date
-----
update      district
set         d_ytd       = d_ytd +
@h_amount,

            @d_street_1 = d_street_1,
            @d_street_2 = d_street_2,
            @d_city     = d_city,
            @d_state    = d_state,
            @d_zip      = d_zip,
            @d_name     = d_name,
            @d_id_local = d_id

where       d_w_id      = @w_id and
            d_id        = @d_id

-----
-- get warehouse data and update year-to-date
-----
update      warehouse
set         w_ytd       = w_ytd +
@h_amount,

            @w_street_1 = w_street_1,
            @w_street_2 = w_street_2,
            @w_city     = w_city,
            @w_state    = w_state,
            @w_zip      = w_zip,
            @w_name     = w_name,
            @w_id_local = w_id

where       w_id        = @w_id

-----
-- create history record
-----
insert into history values ( @c_id_local,
                             @c_d_id,
                             @c_w_id,
                             @d_id_local,

```

```

            @w_id_local,
            @datetime,

            @h_amount,

            @w_name + ' ' + @d_name)
commit tran p

-----
-- return data to client
-----
select      @c_id,
            @c_last,
            @datetime,
            @w_street_1,
            @w_street_2,
            @w_city,
            @w_state,
            @w_zip,
            @w_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')
drop procedure 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 -
@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,

where       @c_id_local = c_id
            c_id         = @c_id and

```

```

            c_d_id       = @c_d_id and

-----
-- 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_id_id     = @d_id_id and

-----
-- 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,
@c_h_amount,
@c_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 = 'sp4tpcc_payment' )
    drop procedure sp4tpcc_payment
go

create proc sp4tpcc_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_credit datetime,
@c_credit_lim numeric(12,2),
@c_balance numeric(12,2),
@c_discount numeric(4,4),
@c_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,
      @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 +
@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,
@h_amount,
@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_d_id        = @c_d_id and

-----
-- 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_id_id = @d_id_id and

-----
-- 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,
@c_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 = 'sp6tpcc_payment' )
    drop procedure sp6tpcc_payment
go

create proc sp6tpcc_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_credit datetime,
@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,
      @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 +
@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,
@h_amount,
@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 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_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_d_id      = @c_d_id and

-----
-- 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_id_id = @d_id_id and

-----
-- 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,

@c_id_local,

@c_w_id_local,

@datetime,

@c_h_amount,

@c_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
```

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 = 'sp1tpcc_stocklevel' )
drop procedure sp1tpcc_stocklevel
go

create proc sp1tpcc_stocklevel @w_id int,
                              @d_id tinyint,
                              @threshold smallint
as

```

```

        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_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
                @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 = '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

```

```

s_quantity < @threshold ol_i_id and

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))

```

```

where stock_id order_line      = @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 123
#define DEF_STARTING_WAREHOUSE 1
#define BUILD_INDEX 1 // build both data and indexes
#define INDEX_ORDER 1 // build indexes before load
#define SCALE_DOWN 0 // build a normal scale database
```

```
#define INDEX_SCRIPT_PATH "scripts"
typedef struct
{
    char *server;
    char *database;
    char *user;
    char *password;

    BOOL tables_all; // set if loading all
    BOOL table_item; // set if loading ITEM
    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;
} TPCC_LDR_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_NAME_LEN 30
#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()Tunable Parameters
```

Appendix C Tunable Parameters

Disabled Windows Services

Alerter
Automatic Updates
ClipBook
Distributed Link Tracking Client
Error Reporting Service
HID Input Service
IMAPI CD-Burning COM Service
Indexing Service
Intersite Messaging
Kerberos Key Distribution Center
License Logging Service
Messenger
Network DDE
Network DDE DSDM
Print Spooler
Remote Registry Service
Secondary Logon
Telnet
Terminal Services Session Directory
Upload Manager
WebClient
Windows Audio
Windows Image Acquisition
Wireless Configuration

Server System Configuration

System Information report written at: 12/08/03 16:51:04

System Name: SQLOLY

[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	SQLOLY		
System Manufacturer	hp		
System Model	server rx8620		
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

BIOS Version/Date	HP 001.002, Not Available
SMBIOS Version	2.3
Windows Directory	C:\windows
System Directory	C:\windows\system32
Boot Device	\Device\HarddiskVolume9
Locale	United States
Hardware Abstraction Layer	Version = "5.2.3790.73 (srv03_qfe.030614-1407)"
User Name	Not Available
Time Zone	Pacific Standard Time
Total Physical Memory	131,005.45 MB
Available Physical Memory	125.18 GB
Total Virtual Memory	254.77 GB
Available Virtual Memory	251.38 GB
Page File Space	126.83 GB
Page File	C:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

Resource	Device
I/O Port 0x0000A000-0x0000BFFF	PCI bus
I/O Port 0x0000A000-0x0000BFFF	Emulex LightPulse PCI Fibre Channel HBA
I/O Port 0x0000A000-0x0000BFFF	PCI bus
I/O Port 0x0000A000-0x0000BFFF	Emulex LightPulse PCI Fibre Channel HBA
I/O Port 0x0000A000-0x0000BFFF	PCI bus
I/O Port 0x0000A000-0x0000BFFF	PCI bus

Memory Address 0xF0000000-0xFFFFFFFF	PCI bus
Memory Address 0xF0000000-0xFFFFFFFF	Smart Array 5300 Controller (Non-Miniport)
Memory Address 0xF0000000-0xFFFFFFFF	PCI bus
Memory Address 0xF0000000-0xFFFFFFFF	Smart Array 5300 Controller (Non-Miniport)
Memory Address 0xF0000000-0xFFFFFFFF	PCI bus
Memory Address 0xF0000000-0xFFFFFFFF	Smart Array 5300 Controller (Non-Miniport)
Memory Address 0xF0000000-0xFFFFFFFF	PCI bus
Memory Address 0xF0000000-0xFFFFFFFF	Smart Array 5300 Controller (Non-Miniport)
Memory Address 0xF0000000-0xFFFFFFFF	PCI bus
Memory Address 0xF0000000-0xFFFFFFFF	Smart Array 5300 Controller (Non-Miniport)
Memory Address 0xF0000000-0xFFFFFFFF	PCI bus
Memory Address 0xF0000000-0xFFFFFFFF	Smart Array 5300 Controller (Non-Miniport)
Memory Address 0xF0000000-0xFFFFFFFF	PCI bus
Memory Address 0xF0000000-0xFFFFFFFF	Smart Array 5300 Controller (Non-Miniport)
Memory Address 0xF0000000-0xFFFFFFFF	PCI bus
Memory Address 0xF0000000-0xFFFFFFFF	Smart Array 5300 Controller (Non-Miniport)

PCI bus
HP Baseboard Management Controller Interface Driver

PCI bus
PCI bus

I/O Port 0x00000000-0x000002FF	PCI bus
I/O Port 0x00000000-0x000002FF	PCI bus
I/O Port 0x00000000-0x000002FF	PCI bus
I/O Port 0x00000000-0x000002FF	PCI bus

Memory Address 0xC0014000-0xC0015FFF	LSI Logic 53C1010-66 Device
--------------------------------------	-----------------------------

Memory Address 0xC0014000-0xC0015FFF Device	LSI Logic 53C1010-66	I/O Port 0x00006000-0x00007FFF Miniport)	Smart Array 5300 Controller (Non- PCI bus	Memory Address 0xC0010000-0xC0011FFF Device	LSI Logic 53C1010-66
Memory Address 0xC0014000-0xC0015FFF Device	LSI Logic 53C1010-66	I/O Port 0x00006000-0x00007FFF Miniport)	Smart Array 5300 Controller (Non- Miniport)	Memory Address 0xC0010000-0xC0011FFF Device	LSI Logic 53C1010-66
I/O Port 0x00002000-0x00003FFF	PCI bus	I/O Port 0x00006000-0x00007FFF Miniport)	Smart Array 5300 Controller (Non- Miniport)	Memory Address 0xC001C000-0xC001C3FF Device	LSI Logic 53C1010-66
I/O Port 0x00002000-0x00003FFF	PCI bus	I/O Port 0x0000E000-0x0000FFFF	PCI bus	Memory Address 0xC001C000-0xC001C3FF Device	LSI Logic 53C1010-66
I/O Port 0x00002000-0x00003FFF	PCI bus	I/O Port 0x0000E000-0x0000FFFF	Smart Array 5300 Controller (Non- Miniport)	Memory Address 0xC001C000-0xC001C3FF Device	LSI Logic 53C1010-66
I/O Port 0x00002000-0x00003FFF	PCI bus	I/O Port 0x0000E000-0x0000FFFF	Smart Array 5300 Controller (Non- Miniport)	Memory Address 0xE0000000-0xEFFFFFFF	PCI bus
Memory Address 0xC0018000-0xC001800F Port (COM5)	HP MP Serial AUX/UPS	I/O Port 0x0000E000-0x0000FFFF	Smart Array 5300 Controller (Non- Miniport)	Memory Address 0xE0000000-0xEFFFFFFF	Smart Array 5300
Memory Address 0xC0018000-0xC001800F Port (COM4)	HP MP Serial AUX/UPS	I/O Port 0x0000E000-0x0000FFFF	PCI bus	Memory Address 0xE0000000-0xEFFFFFFF	PCI bus
Memory Address 0xC0018000-0xC001800F Port (COM3)	HP MP Serial AUX/UPS	I/O Port 0x0000E000-0x0000FFFF	Smart Array 5300 Controller (Non- Miniport)	Memory Address 0xE0000000-0xEFFFFFFF	Smart Array 5300
Memory Address 0xC0020000-0xC0020FFF Fibre Channel Adapter	QLogic QLA23xx PCI	I/O Port 0x0000E000-0x0000FFFF	PCI bus	Memory Address 0xE0000000-0xEFFFFFFF	PCI bus
Memory Address 0xC0020000-0xC0020FFF Fibre Channel Adapter	QLogic QLA23xx PCI	I/O Port 0x0000E000-0x0000FFFF	Smart Array 5300 Controller (Non- Miniport)	Memory Address 0xE0000000-0xEFFFFFFF	Smart Array 5300
Memory Address 0xC0020000-0xC0020FFF Fibre Channel Adapter	QLogic QLA23xx PCI	Memory Address 0xF0200000-0xF023FFFF Controller (Non-Miniport)	Smart Array 5300	Memory Address 0xE0000000-0xEFFFFFFF	Smart Array 5300
Memory Address 0xC0020000-0xC0020FFF Fibre Channel Adapter	QLogic QLA23xx PCI	Memory Address 0xF0200000-0xF023FFFF Controller (Non-Miniport)	Smart Array 5300	Memory Address 0xE0000000-0xEFFFFFFF	PCI bus
I/O Port 0x00000800-0x000008FF	LSI Logic 53C1010-66 Device	Memory Address 0xF0200000-0xF023FFFF Controller (Non-Miniport)	Smart Array 5300	Memory Address 0xE0000000-0xEFFFFFFF	Smart Array 5300
I/O Port 0x00000800-0x000008FF	LSI Logic 53C1010-66 Device	Memory Address 0xF0200000-0xF023FFFF Controller (Non-Miniport)	Smart Array 5300	Memory Address 0xE0000000-0xEFFFFFFF	PCI bus
I/O Port 0x00000800-0x000008FF	LSI Logic 53C1010-66 Device	Memory Address 0xF0200000-0xF023FFFF Controller (Non-Miniport)	Smart Array 5300	Memory Address 0xE0000000-0xEFFFFFFF	Smart Array 5300
Memory Address 0xC001D000-0xC001D3FF Device	LSI Logic 53C1010-66	Memory Address 0xF0200000-0xF023FFFF Controller (Non-Miniport)	Smart Array 5300	Memory Address 0xE0000000-0xEFFFFFFF	PCI bus
Memory Address 0xC001D000-0xC001D3FF Device	LSI Logic 53C1010-66	Memory Address 0xF0200000-0xF023FFFF Controller (Non-Miniport)	Smart Array 5300	Memory Address 0xE0000000-0xEFFFFFFF	PCI bus
Memory Address 0xC001D000-0xC001D3FF Device	LSI Logic 53C1010-66	I/O Port 0x00000700-0x000007FF	LSI Logic 53C1010-66 Device	Memory Address 0xD0000000-0xDFFFFFFF	Emulex LightPulse PCI
Memory Address 0xC0012000-0xC0013FFF Device	LSI Logic 53C1010-66	I/O Port 0x00000700-0x000007FF	LSI Logic 53C1010-66 Device	Memory Address 0xD0000000-0xDFFFFFFF	PCI bus
Memory Address 0xC0012000-0xC0013FFF Device	LSI Logic 53C1010-66	I/O Port 0x00000700-0x000007FF	LSI Logic 53C1010-66 Device	Memory Address 0xD0000000-0xDFFFFFFF	PCI bus
Memory Address 0xC0012000-0xC0013FFF Device	LSI Logic 53C1010-66	Memory Address 0xC8000000-0xCFFFFFFF	PCI bus	Memory Address 0xD0000000-0xDFFFFFFF	Emulex LightPulse PCI
I/O Port 0x00006000-0x00007FFF	PCI bus	Memory Address 0xC8000000-0xCFFFFFFF	PCI bus	Memory Address 0xD0000000-0xDFFFFFFF	Fibre Channel HBA
I/O Port 0x00006000-0x00007FFF	Smart Array 5300 Controller (Non- Miniport)	Memory Address 0xC8000000-0xCFFFFFFF	PCI bus	Memory Address 0xD0000000-0xDFFFFFFF	PCI bus
I/O Port 0x00006000-0x00007FFF	PCI bus	Memory Address 0xC8000000-0xCFFFFFFF	PCI bus	Memory Address 0xD0000000-0xDFFFFFFF	PCI bus
		Memory Address 0xC8000000-0xCFFFFFFF	PCI bus	Memory Address 0xD0000000-0xDFFFFFFF	PCI bus
		Memory Address 0xC0010000-0xC0011FFF Device	LSI Logic 53C1010-66	Memory Address 0xC0019000-0xC0019FFF Processor	HP Management
				Memory Address 0xC0019000-0xC0019FFF Processor	HP Management

Memory Address 0xC0019000-0xC0019FFF Processor	HP Management	I/O Port 0x0000500-0x00005FF I/O Port 0x0000500-0x00005FF	LSI Logic 53C1010-66 Device LSI Logic 53C1010-66 Device	Memory Address 0xE0200000-0xE023FFFF Memory Address 0xE0200000-0xE023FFFF Controller (Non-Miniport)	Smart Array 5300 Smart Array 5300
Memory Address 0xD0001000-0xD00010FF Fibre Channel HBA	Emulex LightPulse PCI	I/O Port 0x00004000-0x00005FFF I/O Port 0x00004000-0x00005FFF	PCI bus Smart Array 5300 Controller (Non-Miniport)	Memory Address 0xE0200000-0xE023FFFF Controller (Non-Miniport)	Smart Array 5300
Memory Address 0xC0016000-0xC0017FFF Device	LSI Logic 53C1010-66	I/O Port 0x00004000-0x00005FFF I/O Port 0x00004000-0x00005FFF	PCI bus Smart Array 5300 Controller (Non-Miniport)	I/O Port 0x00000600-0x000006FF	LSI Logic 53C1010-66 Device
Memory Address 0xC0016000-0xC0017FFF Device	LSI Logic 53C1010-66	I/O Port 0x00004000-0x00005FFF I/O Port 0x00004000-0x00005FFF	PCI bus Smart Array 5300 Controller (Non-Miniport)	I/O Port 0x00000600-0x000006FF	LSI Logic 53C1010-66 Device
Memory Address 0xC0016000-0xC0017FFF Device	LSI Logic 53C1010-66	I/O Port 0x00004000-0x00005FFF I/O Port 0x00004000-0x00005FFF	PCI bus Smart Array 5300 Controller (Non-Miniport)	I/O Port 0x00000600-0x000006FF	LSI Logic 53C1010-66 Device
Memory Address 0xC001A000-0xC001A3FF Device	LSI Logic 53C1010-66	I/O Port 0x0000C000-0x0000DFFF I/O Port 0x0000C000-0x0000DFFF	PCI bus Smart Array 5300 Controller (Non-Miniport)	[DMA]	
Memory Address 0xC001A000-0xC001A3FF Device	LSI Logic 53C1010-66	I/O Port 0x0000C000-0x0000DFFF I/O Port 0x0000C000-0x0000DFFF	PCI bus Smart Array 5300 Controller (Non-Miniport)	Resource Device Status	
Memory Address 0xC001A000-0xC001A3FF Device	LSI Logic 53C1010-66	I/O Port 0x0000C000-0x0000DFFF I/O Port 0x0000C000-0x0000DFFF	PCI bus Smart Array 5300 Controller (Non-Miniport)	[Forced Hardware]	
Memory Address 0xC0000000-0xC7FFFFFFF Gigabit Ethernet #4	PCI bus Broadcom NetXtreme	I/O Port 0x0000C000-0x0000DFFF I/O Port 0x0000C000-0x0000DFFF	PCI bus Smart Array 5300 Controller (Non-Miniport)	Device PNP Device ID	
Memory Address 0xC0000000-0xC7FFFFFFF	PCI bus	I/O Port 0x0000C000-0x0000DFFF	PCI bus	[I/O]	
Memory Address 0xC0000000-0xC7FFFFFFF	PCI bus	I/O Port 0x0000C000-0x0000DFFF	PCI bus	Resource Device Status	
Memory Address 0xC0000000-0xC7FFFFFFF	PCI bus	I/O Port 0x0000C000-0x0000DFFF	Smart Array 5300 Controller (Non-Miniport)	0x00000000-0x000002FF	PCI bus OK
Memory Address 0xC0000000-0xC7FFFFFFF	PCI bus	I/O Port 0x0000C000-0x0000DFFF	Smart Array 5300 Controller (Non-Miniport)	0x00000000-0x000002FF	PCI bus OK
Memory Address 0xC0000000-0xC7FFFFFFF	Broadcom NetXtreme	I/O Port 0x00008000-0x00009FFF I/O Port 0x00008000-0x00009FFF	PCI bus QLogic QLA23xx PCI Fibre Channel Adapter	0x00000000-0x000002FF	PCI bus OK
Memory Address 0xC0000000-0xC7FFFFFFF	PCI bus	I/O Port 0x00008000-0x00009FFF I/O Port 0x00008000-0x00009FFF	PCI bus QLogic QLA23xx PCI Fibre Channel Adapter	0x00000400-0x00000FFF	PCI bus OK
Memory Address 0xC0000000-0xC7FFFFFFF	Broadcom NetXtreme	I/O Port 0x00008000-0x00009FFF I/O Port 0x00008000-0x00009FFF	PCI bus QLogic QLA23xx PCI Fibre Channel Adapter	0x00000500-0x000005FF	LSI Logic 53C1010-66 Device OK
Memory Address 0xC0000000-0xC7FFFFFFF	PCI bus	I/O Port 0x00008000-0x00009FFF I/O Port 0x00008000-0x00009FFF	PCI bus QLogic QLA23xx PCI Fibre Channel Adapter	0x00000500-0x000005FF	LSI Logic 53C1010-66 Device OK
I/O Port 0x00001000-0x00001FFF	PCI bus	I/O Port 0x00008000-0x00009FFF I/O Port 0x00008000-0x00009FFF	PCI bus QLogic QLA23xx PCI Fibre Channel Adapter	0x00000600-0x000006FF	LSI Logic 53C1010-66 Device OK
I/O Port 0x00001000-0x00001FFF	PCI bus	I/O Port 0x00008000-0x00009FFF I/O Port 0x00008000-0x00009FFF	PCI bus QLogic QLA23xx PCI Fibre Channel Adapter	0x00000600-0x000006FF	LSI Logic 53C1010-66 Device OK
I/O Port 0x00001000-0x00001FFF	PCI bus	I/O Port 0x00008000-0x00009FFF I/O Port 0x00008000-0x00009FFF	PCI bus QLogic QLA23xx PCI Fibre Channel Adapter	0x00000700-0x000007FF	LSI Logic 53C1010-66 Device OK
I/O Port 0x00001000-0x00001FFF	PCI bus	I/O Port 0x00008000-0x00009FFF I/O Port 0x00008000-0x00009FFF	PCI bus QLogic QLA23xx PCI Fibre Channel Adapter	0x00000700-0x000007FF	LSI Logic 53C1010-66 Device OK
Memory Address 0xC001B000-0xC001B3FF Device	LSI Logic 53C1010-66	Memory Address 0xE0200000-0xE023FFFF Controller (Non-Miniport)	Smart Array 5300	0x00000800-0x000008FF	LSI Logic 53C1010-66 Device OK
Memory Address 0xC001B000-0xC001B3FF Device	LSI Logic 53C1010-66	Memory Address 0xE0200000-0xE023FFFF Controller (Non-Miniport)	Smart Array 5300	0x00000800-0x000008FF	LSI Logic 53C1010-66 Device OK
Memory Address 0xC001B000-0xC001B3FF Device	LSI Logic 53C1010-66	Memory Address 0xE0200000-0xE023FFFF Controller (Non-Miniport)	Smart Array 5300	0x00000800-0x000008FF	LSI Logic 53C1010-66 Device OK
I/O Port 0x0000500-0x00005FF	LSI Logic 53C1010-66 Device	Memory Address 0xE0200000-0xE023FFFF Controller (Non-Miniport)	Smart Array 5300	0x00000800-0x000008FF	LSI Logic 53C1010-66 Device OK
		Memory Address 0xE0200000-0xE023FFFF Controller (Non-Miniport)	Smart Array 5300		

HP Server rx8620 TPC Benchmark® C Full Disclosure Report April 13, 2004 205

0xF0200000-0xF023FFFF Smart Array 5300 Controller (Non-Miniport)
OK
0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
OK
0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
OK
0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
OK
0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
OK
0xD0001000-0xD00010FF Emulex LightPulse PCI Fibre Channel HBA
OK
0xD0001000-0xD00010FF Emulex LightPulse PCI Fibre Channel HBA
OK

[Components]

[Multimedia]

[Audio Codecs]

CODEC	Manufacturer	Description	Status	File	Version
		Size	Creation Date		
c:\windows\system32\imaadp32.acm	Microsoft Corporation				
	OK	C:\windows\system32\IMAADP32.ACM			
5.2.3790.0 (srv03_rtm.030324-2048)		55.00 KB (56,320 bytes)			
10/16/2003 11:30 AM					
c:\windows\system32\msg711.acm	Microsoft Corporation				
	OK	C:\windows\system32\MSG711.ACM			
5.2.3790.0 (srv03_rtm.030324-2048)		33.00 KB (33,792 bytes)			
10/16/2003 11:30 AM					
c:\windows\system32\msgsm32.acm	Microsoft Corporation				
	OK	C:\windows\system32\MSGSM32.ACM			
5.2.3790.0 (srv03_rtm.030324-2048)		66.50 KB (68,096 bytes)			
10/16/2003 11:30 AM					
c:\windows\system32\msadp32.acm	Microsoft Corporation				
	OK	C:\windows\system32\MSADP32.ACM			
5.2.3790.0 (srv03_rtm.030324-2048)		49.00 KB (50,176 bytes)			
10/16/2003 11:30 AM					
c:\windows\system32\tssoft32.acm	DSP GROUP, INC.				
	OK	C:\windows\system32\TSSOFT32.ACM	1.01		
29.00 KB (29,696 bytes)		10/16/2003 11:30 AM			

[Video Codecs]

CODEC	Manufacturer	Description	Status	File	Version
-------	--------------	-------------	--------	------	---------

c:\windows\system32\msvidc32.dll Microsoft Corporation
OK
C:\windows\system32\MSVIDC32.DLL
5.2.3790.0 (srv03_rtm.030324-2048) 67.00 KB (68,608 bytes)
c:\windows\system32\msrle32.dll Microsoft Corporation
OK
C:\windows\system32\MSRLE32.DLL
5.2.3790.0 (srv03_rtm.030324-2048) 24.50 KB (25,088 bytes)
9/18/2003 11:53 AM

[CD-ROM]

Item	Value
Drive	D:
Description	CD-ROM Drive
Media Loaded	No
Media Type	CD-ROM
Name	HP DVD-ROM 305 SCSI CdRom Device
Manufacturer	(Standard CD-ROM drives)
Status	OK
Transfer Rate	Not Available
SCSI Target ID	2
PNP Device ID	SCSI\CDROM&VEN_HP&PROD_DVD-ROM_305&REV_1.01\6&181172F5&0&020
Driver	c:\windows\system32\drivers\cdrom.sys (5.2.3790.0 (srv03_rtm.030324-2048), 143.50 KB (146,944 bytes), 10/16/2003 11:30 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	Not Available
Product Type	Intel(R) PRO/1000 T Server Adapter
Installed	Yes
PNP Device ID	Not Available
Last Reset	12/4/2003 11:41 AM
Index	1
Service Name	E1000
IP Address	Not Available
IP Subnet	Not Available
Default IP Gateway	Not Available
DHCP Enabled	Yes
DHCP Server	Not Available
DHCP Lease Expires	Not Available
DHCP Lease Obtained	Not Available
MAC Address	Not Available

Name [00000002] RAS Async Adapter

Adapter Type	Not Available
Product Type	RAS Async Adapter
Installed	Yes
PNP Device ID	Not Available
Last Reset	12/4/2003 11:41 AM
Index	2
Service Name	AsyncMac
IP Address	Not Available
IP Subnet	Not Available
Default IP Gateway	Not Available
DHCP Enabled	No
DHCP Server	Not Available
DHCP Lease Expires	Not Available
DHCP Lease Obtained	Not Available
MAC Address	Not Available

Name [00000003] WAN Miniport (L2TP)

Adapter Type	Not Available
Product Type	WAN Miniport (L2TP)
Installed	Yes
PNP Device ID	ROOT\MS_L2TPMINI\PORT\0000
Last Reset	12/4/2003 11:41 AM
Index	3
Service Name	Rasl2tp
IP Address	Not Available
IP Subnet	Not Available
Default IP Gateway	Not Available
DHCP Enabled	No

DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\rasl2tp.sys (5.2.3790.0
(srv03_rtm.030324-2048), 179.50 KB (183,808 bytes), 10/16/2003 11:30 AM)

Name [00000004] WAN Miniport (PPTP)
Adapter TypeWide Area Network (WAN)
Product TypeWAN Miniport (PPTP)
Installed Yes
PNP Device ID ROOT\MS_PPTPMINIPORT\0000
Last Reset 12/4/2003 11:41 AM
Index 4

Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30
Driver c:\windows\system32\drivers\raspptp.sys (5.2.3790.0
(srv03_rtm.030324-2048), 177.50 KB (181,760 bytes), 10/16/2003 11:30 AM)

Name [00000005] WAN Miniport (PPPOE)
Adapter TypeWide Area Network (WAN)
Product TypeWAN Miniport (PPPOE)
Installed Yes
PNP Device ID ROOT\MS_PPPOEMINIPORT\0000
Last Reset 12/4/2003 11:41 AM
Index 5

Service Name RasPppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30
Driver c:\windows\system32\drivers\raspppoe.sys (5.2.3790.0
(srv03_rtm.030324-2048), 115.50 KB (118,272 bytes), 9/18/2003 11:53 AM)

Name [00000006] Direct Parallel
Adapter TypeNot Available
Product TypeDirect Parallel
Installed Yes

PNP Device ID ROOT\MS_PTMINIPORT\0000
Last Reset 12/4/2003 11:41 AM
Index 6
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\raspti.sys (5.2.3790.0
(srv03_rtm.030324-2048), 49.50 KB (50,688 bytes), 9/18/2003 11:53 AM)

Name [00000007] WAN Miniport (IP)
Adapter TypeNot Available
Product TypeWAN Miniport (IP)
Installed Yes
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 12/4/2003 11:41 AM
Index 7

Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\ndiswan.sys (5.2.3790.0
(srv03_rtm.030324-2048), 250.00 KB (256,000 bytes), 10/16/2003 11:30 AM)

Name [00000008] Broadcom NetXtreme Gigabit Ethernet
Adapter TypeNot Available
Product TypeBroadcom NetXtreme Gigabit Ethernet
Installed Yes
PNP Device ID Not Available
Last Reset 12/4/2003 11:41 AM
Index 8

Service Name b57nd
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled Yes
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000009] Broadcom NetXtreme Gigabit Ethernet

Product TypeBroadcom NetXtreme Gigabit Ethernet
Adapter TypeBroadcom NetXtreme Gigabit Ethernet
Installed Yes
PNP Device ID PCI\VEN_14E4&DEV_1645&SUBSYS_1300103C&REV_15\5&14C294CC&0&08

Last Reset 12/4/2003 11:41 AM
Index 9
Service Name b57nd
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 00:30:6E:49:42:AB
Memory Address 0xC0000000-0xC7FFFFFFF
IRQ Channel IRQ 340
Driver c:\windows\system32\drivers\b57xp64.sys (7.15.0.0 built by:
WinDDK, 336.13 KB (344,192 bytes), 9/18/2003 11:55 AM)

Name [00000010] Broadcom NetXtreme Gigabit Ethernet
Adapter TypeEthernet 802.3
Product TypeBroadcom NetXtreme Gigabit Ethernet
Installed Yes
PNP Device ID PCI\VEN_14E4&DEV_1645&SUBSYS_1300103C&REV_15\5&194FB0DF&0&08

Last Reset 12/4/2003 11:41 AM
Index 10
Service Name b57nd
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 00:30:6E:49:42:33
Memory Address 0xC0000000-0xC7FFFFFFF
IRQ Channel IRQ 236
Driver c:\windows\system32\drivers\b57xp64.sys (7.15.0.0 built by:
WinDDK, 336.13 KB (344,192 bytes), 9/18/2003 11:55 AM)

Name [00000011] Broadcom NetXtreme Gigabit Ethernet
Adapter TypeEthernet 802.3
Product TypeBroadcom NetXtreme Gigabit Ethernet
Installed Yes
PNP Device ID PCI\VEN_14E4&DEV_1645&SUBSYS_1300103C&REV_15\5&5CDAB618&0&08

Last Reset 12/4/2003 11:41 AM
Index 11

Service Name b57nd
 IP Address 10.193.24.183
 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 12/16/2003 11:48 AM
 DHCP Lease Obtained 12/8/2003 11:48 AM
 MAC Address 00:30:6E:49:62:E8
 Memory Address 0xC0000000-0xC7FFFFFFF
 IRQ Channel IRQ 28
 Driver c:\windows\system32\drivers\b57xp64.sys (7.15.0.0 built by: WinDDK, 336.13 KB (344,192 bytes), 9/18/2003 11:55 AM)

Name [00000012] Intel(R) PRO/1000 T Server Adapter
 Adapter Type Not Available
 Product Type Intel(R) PRO/1000 T Server Adapter
 Installed Yes
 PNP Device ID Not Available
 Last Reset 12/4/2003 11:41 AM
 Index 12
 Service Name E1000
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled Yes
 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

Name	Connectionless Service	Guarantees Delivery	Guarantees Sequencing	Maximum Address Size	Maximum Message Size	Message Oriented	Minimum Address Size	Pseudo Stream Oriented	Supports Broadcasting	Supports Connect Data	Supports Disconnect Data	Supports Encryption	Supports Expedited Data	Supports Graceful Closing	Supports Guaranteed Bandwidth	Supports Multicasting
MSAFD Tcpip [TCP/IP]	Yes	No	No	16 bytes	63.93 KB (65,467 bytes)	Yes	16 bytes	No	Yes	No	No	No	No	No	No	Yes

Name	Connectionless Service	Guarantees Delivery	Guarantees Sequencing	Maximum Address Size	Maximum Message Size	Message Oriented	Minimum Address Size	Pseudo Stream Oriented	Supports Broadcasting	Supports Connect Data	Supports Disconnect Data	Supports Encryption	Supports Expedited Data	Supports Graceful Closing	Supports Guaranteed Bandwidth	Supports Multicasting
RSVP UDP Service Provider	Yes	No	No	16 bytes	63.93 KB (65,467 bytes)	Yes	16 bytes	No	Yes	No	No	Yes	No	No	No	Yes

Name	Connectionless Service	Guarantees Delivery	Guarantees Sequencing	Maximum Address Size	Maximum Message Size	Message Oriented	Minimum Address Size	Pseudo Stream Oriented	Supports Broadcasting	Supports Connect Data	Supports Disconnect Data	Supports Encryption	Supports Expedited Data	Supports Graceful Closing	Supports Guaranteed Bandwidth	Supports Multicasting
RSVP TCP Service Provider	No	Yes	Yes	16 bytes	0 bytes	No	16 bytes	No	No	No	No	No	Yes	Yes	No	No

[WinSock]	Item	Value
	File	c:\windows\system32\wssock32.dll
	Size	23.00 KB (23,552 bytes)
	Version	5.2.3790.0 (srv03_rtm.030324-2048)

[Ports]

[Serial]

Item	Value
Name	HP MP Serial AUX/UPS Port (COM5)
Status	OK
PNP Device ID	PCI\VEN_103C&DEV_1290&SUBSYS_1291103C&REV_01\5&5C DAB61&0&00
Maximum Input Buffer Size	0
Maximum Output Buffer Size	No
Settable Baud Rate	Yes
Settable Data Bits	Yes
Settable Flow Control	Yes
Settable Parity	Yes
Settable Parity Check	Yes
Settable Stop Bits	Yes
Settable RLSD	Yes
Supports RLSD	Yes
Supports 16 Bit Mode	No
Supports Special Characters	No
Baud Rate	9600
Bits/Byte	8
Stop Bits	1
Parity	None
Busy	No
Abort Read/Write on Error	No
Binary Mode Enabled	Yes
Continue XMit on XOff	No
CTS Outflow Control	No
Discard NULL Bytes	No
DSR Outflow Control	0
DSR Sensitivity	0
DTR Flow Control Type	Enable
EOF Character	0
Error Replace Character	0
Error Replacement Enabled	No
Event Character	0
Parity Check Enabled	No
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
 Memory Address 0xC0018000-0xC001800F
 IRQ Channel IRQ 29
 Driver c:\windows\system32\drivers\hmpser.sys (5.0.3663.16,
 154.00 KB (157,696 bytes), 9/18/2003 11:55 AM)

 Name HP MP Serial AUX/UPS Port (COM4)
 Status OK
 PNP Device ID
 PCI\VEN_103C&DEV_1290&SUBSYS_1291103C&REV_01\5&194
 FB0DF&0&00
 Maximum Input Buffer Size0
 Maximum Output Buffer Size No
 Settable Baud Rate Yes
 Settable Data Bits Yes
 Settable Flow Control Yes
 Settable Parity Yes
 Settable Parity Check Yes
 Settable Stop Bits Yes
 Settable RLSD Yes
 Supports RLSD Yes
 Supports 16 Bit Mode No
 Supports Special Characters No
 Baud Rate 9600
 Bits/Byte 8
 Stop Bits 1
 Parity None
 Busy No
 Abort Read/Write on Error No
 Binary Mode Enabled Yes
 Continue XMit on XOff No
 CTS Outflow Control No
 Discard NULL Bytes No
 DSR Outflow Control 0
 DSR Sensitivity 0
 DTR Flow Control Type Enable
 EOF Character 0
 Error Replace Character 0
 Error Replacement Enabled No
 Event Character 0
 Parity Check Enabled No
 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
 Memory Address 0xC0018000-0xC001800F
 IRQ Channel IRQ 237

Driver c:\windows\system32\drivers\hmpser.sys (5.0.3663.16,
 154.00 KB (157,696 bytes), 9/18/2003 11:55 AM)
 Name HP MP Serial AUX/UPS Port (COM3)
 Status OK
 PNP Device ID
 PCI\VEN_103C&DEV_1290&SUBSYS_1291103C&REV_01\5&14C
 294CC&0&00
 Maximum Input Buffer Size0
 Maximum Output Buffer Size No
 Settable Baud Rate Yes
 Settable Data Bits Yes
 Settable Flow Control Yes
 Settable Parity Yes
 Settable Parity Check Yes
 Settable Stop Bits Yes
 Settable RLSD Yes
 Supports RLSD Yes
 Supports 16 Bit Mode No
 Supports Special Characters No
 Baud Rate 9600
 Bits/Byte 8
 Stop Bits 1
 Parity None
 Busy No
 Abort Read/Write on Error No
 Binary Mode Enabled Yes
 Continue XMit on XOff No
 CTS Outflow Control No
 Discard NULL Bytes No
 DSR Outflow Control 0
 DSR Sensitivity 0
 DTR Flow Control Type Enable
 EOF Character 0
 Error Replace Character 0
 Error Replacement Enabled No
 Event Character 0
 Parity Check Enabled No
 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
 Memory Address 0xC0018000-0xC001800F
 IRQ Channel IRQ 341
 Driver c:\windows\system32\drivers\hmpser.sys (5.0.3663.16,
 154.00 KB (157,696 bytes), 9/18/2003 11:55 AM)

 [Parallel]

 Item Value

[Storage]

[Drives]

Item	Value
Drive	C:
Description	Local Fixed Disk
Compressed	No
File System	NTFS
Size	16.35 GB (17,551,495,168 bytes)
Free Space	9.90 GB (10,633,256,960 bytes)
Volume Name	18GB_BOOT
Volume Serial Number	E87A79AD
Drive	D:
Description	CD-ROM Disc
Drive	G:
Description	Local Fixed Disk
Compressed	No
File System	NTFS
Size	16.85 GB (18,095,611,904 bytes)
Free Space	16.50 GB (17,714,073,600 bytes)
Volume Name	New Volume
Volume Serial Number	20DD9082
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	S:
Description	Network Connection
Provider Name	\\ssd\\lab\\d\$
Drive	T:
Description	Local Fixed Disk
Compressed	No
File System	NTFS
Size	97.66 GB (104,857,595,904 bytes)
Free Space	95.79 GB (102,851,543,040 bytes)
Volume Name	BackupLog
Volume Serial Number	2C55398A
Drive	Z:
Description	Local Fixed Disk
Compressed	No
File System	FAT32

Size 16.59 GB (17,815,453,696 bytes)
Free Space 3.21 GB (3,443,376,128 bytes)
Volume Name
Volume Serial Number 4414B961

[Disks]

Item	Value
Description	\\.\PHYSICALDRIVE0
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	No
Media Type	Fixed hard disk
Partitions	Not Available
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	149.42 GB (160,434,086,400 bytes)
Total Cylinders	19,505
Total Sectors	313,347,825
Total Tracks	4,973,775
Tracks/Cylinder	255
Partition	Disk #0, Partition #0
Partition Size	96.08 GB (103,164,149,760 bytes)
Partition Starting Offset	136,314,880 bytes
Partition	Disk #0, Partition #1
Partition Size	53.21 GB (57,131,663,360 bytes)
Partition Starting Offset	103,300,464,640 bytes

Description	\\.\PHYSICALDRIVE1
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	No
Media Type	Fixed hard disk
Partitions	Not Available
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	400.12 GB (429,622,824,960 bytes)
Total Cylinders	52,232
Total Sectors	839,107,080
Total Tracks	13,319,160
Tracks/Cylinder	255
Partition	Disk #1, Partition #0
Partition Size	133.79 GB (143,654,912,000 bytes)
Partition Starting Offset	136,314,880 bytes

Partition Size 133.79 GB (143,654,912,000 bytes)
Partition Starting Offset 143,791,226,880 bytes

Description	\\.\PHYSICALDRIVE4
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	No
Media Type	Fixed hard disk
Partitions	Not Available
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	149.42 GB (160,434,086,400 bytes)
Total Cylinders	19,505
Total Sectors	313,347,825
Total Tracks	4,973,775
Tracks/Cylinder	255
Partition	Disk #4, Partition #0
Partition Size	96.08 GB (103,164,149,760 bytes)
Partition Starting Offset	136,314,880 bytes
Partition	Disk #4, Partition #1
Partition Size	53.21 GB (57,131,663,360 bytes)
Partition Starting Offset	103,300,464,640 bytes

Description	\\.\PHYSICALDRIVE5
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	No
Media Type	Fixed hard disk
Partitions	Not Available
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	400.12 GB (429,622,824,960 bytes)
Total Cylinders	52,232
Total Sectors	839,107,080
Total Tracks	13,319,160
Tracks/Cylinder	255
Partition	Disk #5, Partition #0
Partition Size	133.79 GB (143,654,912,000 bytes)
Partition Starting Offset	136,314,880 bytes
Partition	Disk #5, Partition #1
Partition Size	133.79 GB (143,654,912,000 bytes)
Partition Starting Offset	143,791,226,880 bytes

Description \\.\PHYSICALDRIVE12
Manufacturer Not Available

Bytes/Sector	512
Media Loaded	No
Media Type	Fixed hard disk
Partitions	Not Available
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	149.42 GB (160,434,086,400 bytes)
Total Cylinders	19,505
Total Sectors	313,347,825
Total Tracks	4,973,775
Tracks/Cylinder	255
Partition	Disk #12, Partition #0
Partition Size	96.08 GB (103,164,149,760 bytes)
Partition Starting Offset	136,314,880 bytes
Partition	Disk #12, Partition #1
Partition Size	53.21 GB (57,131,663,360 bytes)
Partition Starting Offset	103,300,464,640 bytes

Description	\\.\PHYSICALDRIVE13
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	No
Media Type	Fixed hard disk
Partitions	Not Available
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	400.12 GB (429,622,824,960 bytes)
Total Cylinders	52,232
Total Sectors	839,107,080
Total Tracks	13,319,160
Tracks/Cylinder	255
Partition	Disk #13, Partition #0
Partition Size	133.79 GB (143,654,912,000 bytes)
Partition Starting Offset	136,314,880 bytes
Partition	Disk #13, Partition #1
Partition Size	133.79 GB (143,654,912,000 bytes)
Partition Starting Offset	143,791,226,880 bytes

Description	\\.\PHYSICALDRIVE24
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	No
Media Type	Fixed hard disk
Partitions	Not Available
SCSI Bus	Not Available

SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 149.42 GB (160,434,086,400 bytes)
 Total Cylinders 19,505
 Total Sectors 313,347,825
 Total Tracks 4,973,775
 Tracks/Cylinder 255
 Partition Disk #24, Partition #0
 Partition Size 96.08 GB (103,164,149,760 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #24, Partition #1
 Partition Size 53.21 GB (57,131,663,360 bytes)
 Partition Starting Offset 103,300,464,640 bytes

Description \\.\PHYSICALDRIVE25
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 400.12 GB (429,622,824,960 bytes)
 Total Cylinders 52,232
 Total Sectors 839,107,080
 Total Tracks 13,319,160
 Tracks/Cylinder 255
 Partition Disk #25, Partition #0
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #25, Partition #1
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 143,791,226,880 bytes

Description \\.\PHYSICALDRIVE8
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63

Total Cylinders 19,505
 Total Sectors 313,347,825
 Total Tracks 4,973,775
 Tracks/Cylinder 255
 Partition Disk #8, Partition #0
 Partition Size 96.08 GB (103,164,149,760 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #8, Partition #1
 Partition Size 53.21 GB (57,131,663,360 bytes)
 Partition Starting Offset 103,300,464,640 bytes

Description \\.\PHYSICALDRIVE9
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 400.12 GB (429,622,824,960 bytes)
 Total Cylinders 52,232
 Total Sectors 839,107,080
 Total Tracks 13,319,160
 Tracks/Cylinder 255
 Partition Disk #9, Partition #0
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #9, Partition #1
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 143,791,226,880 bytes

Description \\.\PHYSICALDRIVE20
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 149.42 GB (160,434,086,400 bytes)
 Total Cylinders 19,505
 Total Sectors 313,347,825
 Total Tracks 4,973,775
 Tracks/Cylinder 255
 Partition Disk #20, Partition #0

Partition Starting Offset 103,300,464,640 bytes
 Partition Disk #20, Partition #1
 Partition Size 53.21 GB (57,131,663,360 bytes)
 Partition Starting Offset 103,300,464,640 bytes

Description \\.\PHYSICALDRIVE21
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 400.12 GB (429,622,824,960 bytes)
 Total Cylinders 52,232
 Total Sectors 839,107,080
 Total Tracks 13,319,160
 Tracks/Cylinder 255
 Partition Disk #21, Partition #0
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #21, Partition #1
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 143,791,226,880 bytes

Description \\.\PHYSICALDRIVE30
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 149.42 GB (160,434,086,400 bytes)
 Total Cylinders 19,505
 Total Sectors 313,347,825
 Total Tracks 4,973,775
 Tracks/Cylinder 255
 Partition Disk #30, Partition #0
 Partition Size 96.08 GB (103,164,149,760 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #30, Partition #1
 Partition Size 53.21 GB (57,131,663,360 bytes)
 Partition Starting Offset 103,300,464,640 bytes

Description \\.\PHYSICALDRIVE31
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 400.12 GB (429,622,824,960 bytes)
 Total Cylinders 52,232
 Total Sectors 839,107,080
 Total Tracks 13,319,160
 Tracks/Cylinder 255
 Partition Disk #31, Partition #0
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #31, Partition #1
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 143,791,226,880 bytes

Description \\.\PHYSICALDRIVE10
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 149.42 GB (160,434,086,400 bytes)
 Total Cylinders 19,505
 Total Sectors 313,347,825
 Total Tracks 4,973,775
 Tracks/Cylinder 255
 Partition Disk #10, Partition #0
 Partition Size 96.08 GB (103,164,149,760 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #10, Partition #1
 Partition Size 53.21 GB (57,131,663,360 bytes)
 Partition Starting Offset 103,300,464,640 bytes

Description \\.\PHYSICALDRIVE11
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512

Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 400.12 GB (429,622,824,960 bytes)
 Total Cylinders 52,232
 Total Sectors 839,107,080
 Total Tracks 13,319,160
 Tracks/Cylinder 255
 Partition Disk #11, Partition #0
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #11, Partition #1
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 143,791,226,880 bytes

Description \\.\PHYSICALDRIVE16
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 149.42 GB (160,434,086,400 bytes)
 Total Cylinders 19,505
 Total Sectors 313,347,825
 Total Tracks 4,973,775
 Tracks/Cylinder 255
 Partition Disk #16, Partition #0
 Partition Size 96.08 GB (103,164,149,760 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #16, Partition #1
 Partition Size 53.21 GB (57,131,663,360 bytes)
 Partition Starting Offset 103,300,464,640 bytes

Description \\.\PHYSICALDRIVE17
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available

Sectors/Track Not Available
 Size 400.12 GB (429,622,824,960 bytes)
 Total Cylinders 52,232
 Total Sectors 839,107,080
 Total Tracks 13,319,160
 Tracks/Cylinder 255
 Partition Disk #17, Partition #0
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #17, Partition #1
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 143,791,226,880 bytes

Description \\.\PHYSICALDRIVE6
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 149.42 GB (160,434,086,400 bytes)
 Total Cylinders 19,505
 Total Sectors 313,347,825
 Total Tracks 4,973,775
 Tracks/Cylinder 255
 Partition Disk #6, Partition #0
 Partition Size 96.08 GB (103,164,149,760 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #6, Partition #1
 Partition Size 53.21 GB (57,131,663,360 bytes)
 Partition Starting Offset 103,300,464,640 bytes

Description \\.\PHYSICALDRIVE7
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 400.12 GB (429,622,824,960 bytes)
 Total Cylinders 52,232
 Total Sectors 839,107,080
 Total Tracks 13,319,160

Tracks/Cylinder 255
 Partition Disk #7, Partition #0
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #7, Partition #1
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 143,791,226,880 bytes

Description \\.\PHYSICALDRIVE18
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 149.42 GB (160,434,086,400 bytes)
 Total Cylinders 19,505
 Total Sectors 313,347,825
 Total Tracks 4,973,775

Tracks/Cylinder 255
 Partition Disk #18, Partition #0
 Partition Size 96.08 GB (103,164,149,760 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #18, Partition #1
 Partition Size 53.21 GB (57,131,663,360 bytes)
 Partition Starting Offset 103,300,464,640 bytes

Description \\.\PHYSICALDRIVE19
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 400.12 GB (429,622,824,960 bytes)
 Total Cylinders 52,232
 Total Sectors 839,107,080
 Total Tracks 13,319,160
 Tracks/Cylinder 255
 Partition Disk #19, Partition #0
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 136,314,880 bytes

Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 143,791,226,880 bytes

Description \\.\PHYSICALDRIVE14
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 149.42 GB (160,434,086,400 bytes)
 Total Cylinders 19,505
 Total Sectors 313,347,825
 Total Tracks 4,973,775
 Tracks/Cylinder 255
 Partition Disk #14, Partition #0
 Partition Size 96.08 GB (103,164,149,760 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #14, Partition #1
 Partition Size 53.21 GB (57,131,663,360 bytes)
 Partition Starting Offset 103,300,464,640 bytes

Description \\.\PHYSICALDRIVE15
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 400.12 GB (429,622,824,960 bytes)
 Total Cylinders 52,232
 Total Sectors 839,107,080
 Total Tracks 13,319,160
 Tracks/Cylinder 255
 Partition Disk #15, Partition #0
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #15, Partition #1
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 143,791,226,880 bytes

Description \\.\PHYSICALDRIVE2
 Manufacturer Not Available

Bytes/Sector 512 Available
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 149.42 GB (160,434,086,400 bytes)
 Total Cylinders 19,505
 Total Sectors 313,347,825
 Total Tracks 4,973,775
 Tracks/Cylinder 255
 Partition Disk #2, Partition #0
 Partition Size 96.08 GB (103,164,149,760 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #2, Partition #1
 Partition Size 53.21 GB (57,131,663,360 bytes)
 Partition Starting Offset 103,300,464,640 bytes

Description \\.\PHYSICALDRIVE3
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 400.12 GB (429,622,824,960 bytes)
 Total Cylinders 52,232
 Total Sectors 839,107,080
 Total Tracks 13,319,160
 Tracks/Cylinder 255
 Partition Disk #3, Partition #0
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #3, Partition #1
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 143,791,226,880 bytes

Description \\.\PHYSICALDRIVE26
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available

SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 149.42 GB (160,434,086,400 bytes)
 Total Cylinders 19,505
 Total Sectors 313,347,825
 Total Tracks 4,973,775
 Tracks/Cylinder 255
 Partition Disk #26, Partition #0
 Partition Size 96.08 GB (103,164,149,760 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #26, Partition #1
 Partition Size 53.21 GB (57,131,663,360 bytes)
 Partition Starting Offset 103,300,464,640 bytes

Description \\.\PHYSICALDRIVE27
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 400.12 GB (429,622,824,960 bytes)
 Total Cylinders 52,232
 Total Sectors 839,107,080
 Total Tracks 13,319,160
 Tracks/Cylinder 255
 Partition Disk #27, Partition #0
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #27, Partition #1
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 143,791,226,880 bytes

Description \\.\PHYSICALDRIVE28
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63

Total Cylinders 19,505
 Total Sectors 313,347,825
 Total Tracks 4,973,775
 Tracks/Cylinder 255
 Partition Disk #28, Partition #0
 Partition Size 96.08 GB (103,164,149,760 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #28, Partition #1
 Partition Size 53.21 GB (57,131,663,360 bytes)
 Partition Starting Offset 103,300,464,640 bytes

Description \\.\PHYSICALDRIVE29
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 400.12 GB (429,622,824,960 bytes)
 Total Cylinders 52,232
 Total Sectors 839,107,080
 Total Tracks 13,319,160
 Tracks/Cylinder 255
 Partition Disk #29, Partition #0
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #29, Partition #1
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 143,791,226,880 bytes

Description \\.\PHYSICALDRIVE22
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 149.42 GB (160,434,086,400 bytes)
 Total Cylinders 19,505
 Total Sectors 313,347,825
 Total Tracks 4,973,775
 Tracks/Cylinder 255
 Partition Disk #22, Partition #0

Partition Starting Offset 103,300,464,640 bytes
 Partition Disk #22, Partition #1
 Partition Size 53.21 GB (57,131,663,360 bytes)
 Partition Starting Offset 103,300,464,640 bytes

Description \\.\PHYSICALDRIVE23
 Manufacturer Not Available
 Model Not Available
 Bytes/Sector 512
 Media Loaded No
 Media Type Fixed hard disk
 Partitions Not Available
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 400.12 GB (429,622,824,960 bytes)
 Total Cylinders 52,232
 Total Sectors 839,107,080
 Total Tracks 13,319,160
 Tracks/Cylinder 255
 Partition Disk #23, Partition #0
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 136,314,880 bytes
 Partition Disk #23, Partition #1
 Partition Size 133.79 GB (143,654,912,000 bytes)
 Partition Starting Offset 143,791,226,880 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model COMPAQ MSA1000 VOLUME SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 1
 SCSI Logical Unit 1
 SCSI Port 16
 SCSI Target ID 124
 Sectors/Track 63
 Size 302.73 GB (325,054,840,320 bytes)
 Total Cylinders 39,519
 Total Sectors 634,872,735
 Total Tracks 10,077,345
 Tracks/Cylinder 255
 Partition Disk #34, Partition #0
 Partition Size 302.73 GB (325,054,808,064 bytes)
 Partition Starting Offset 32,256 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 1
SCSI Logical Unit 2
SCSI Port 16
SCSI Target ID 124
Sectors/Track 63
Size 517.66 GB (555,831,521,280 bytes)
Total Cylinders 67,576
Total Sectors 1,085,608,440
Total Tracks 17,231,880
Tracks/Cylinder 255
Partition Disk #35, Partition #0
Partition Size 97.66 GB (104,857,600,000 bytes)
Partition Starting Offset 134,235,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 1
SCSI Logical Unit 1
SCSI Port 17
SCSI Target ID 124
Sectors/Track 63
Size 302.73 GB (325,054,840,320 bytes)
Total Cylinders 39,519
Total Sectors 634,872,735
Total Tracks 10,077,345
Tracks/Cylinder 255
Partition Disk #36, Partition #0
Partition Size 302.73 GB (325,054,808,064 bytes)
Partition Starting Offset 32,256 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 1
SCSI Logical Unit 2
SCSI Port 17
SCSI Target ID 124
Sectors/Track 63
Size 517.66 GB (555,831,521,280 bytes)

Total Sectors 1,085,608,440
Total Tracks 17,231,880
Tracks/Cylinder 255

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 4
SCSI Target ID 6
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 #32, Partition #0
Partition Size 100.00 MB (104,857,600 bytes)
Partition Starting Offset 17,408 bytes
Partition Disk #32, Partition #1
Partition Size 400.00 MB (419,430,400 bytes)
Partition Starting Offset 104,875,008 bytes
Partition Disk #32, Partition #2
Partition Size 16.35 GB (17,551,497,728 bytes)
Partition Starting Offset 658,523,136 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 12
SCSI Target ID 6
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 #33, Partition #0
Partition Size 345.12 MB (361,880,064 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #33, Partition #1
Partition Size 16.60 GB (17,824,181,760 bytes)

Partition Starting Offset 18,317,698,560 bytes
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_1300103C&REV_01\5&5CDAB61&0&10
I/O Port	0x00000500-0x000005FF
Memory Address	0xC001A000-0xC001A3FF
Memory Address	0xC0010000-0xC0011FFF
IRQ Channel	IRQ 24
Driver	c:\windows\system32\drivers\sym_u3.sys (5.2.3673.0 (_020828-1450), 66.00 KB (67,584 bytes), 10/16/2003 11:30 AM)
Name	LSI Logic 53C1010-66 Device
Manufacturer	LSI Logic Inc.
Status	OK
PNP Device ID	PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&5CDAB61&0&11
I/O Port	0x00000600-0x000006FF
Memory Address	0xC001B000-0xC001B3FF
Memory Address	0xC0012000-0xC0013FFF
IRQ Channel	IRQ 25
Driver	c:\windows\system32\drivers\sym_u3.sys (5.2.3673.0 (_020828-1450), 66.00 KB (67,584 bytes), 10/16/2003 11:30 AM)
Name	LSI Logic 53C1010-66 Device
Manufacturer	LSI Logic Inc.
Status	OK
PNP Device ID	PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&5CDAB61&0&18
I/O Port	0x00000700-0x000007FF
Memory Address	0xC001C000-0xC001C3FF
Memory Address	0xC0014000-0xC0015FFF
IRQ Channel	IRQ 26
Driver	c:\windows\system32\drivers\sym_u3.sys (5.2.3673.0 (_020828-1450), 66.00 KB (67,584 bytes), 10/16/2003 11:30 AM)
Name	LSI Logic 53C1010-66 Device
Manufacturer	LSI Logic Inc.
Status	OK
PNP Device ID	PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&5CDAB61&0&19
I/O Port	0x00000800-0x000008FF

Memory Address 0xC001D000-0xC001D3FF
Memory Address 0xC0016000-0xC0017FFF
IRQ Channel IRQ 27
Driver c:\windows\system32\drivers\sym_u3.sys (5.2.3673.0
(_020828-1450), 66.00 KB (67,584 bytes), 10/16/2003 11:30 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&28C
A0570&0&08
Memory Address 0xE0200000-0xE023FFFF
Memory Address 0xE0000000-0xEFFFFFFF
I/O Port 0x00004000-0x00005FFF
IRQ Channel IRQ 54
Driver c:\windows\system32\drivers\hpcissb.sys (5.6.2.64 built by:
WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&1F3
4A452&0&08
Memory Address 0xF0200000-0xF023FFFF
Memory Address 0xF0000000-0xFFFFFFFF
I/O Port 0x00006000-0x00007FFF
IRQ Channel IRQ 65
Driver c:\windows\system32\drivers\hpcissb.sys (5.6.2.64 built by:
WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&9C1
02BF&0&08
I/O Port 0x00008000-0x00009FFF
Memory Address 0xC0020000-0xC0020FFF
IRQ Channel IRQ 76
Driver c:\windows\system32\drivers\ql2300.sys (8.2.2.10 (W64 VI),
690.75 KB (707,328 bytes), 4/16/2003 7:45 PM)

Name Emulex LightPulse PCI Fibre Channel HBA
Manufacturer Emulex
Status OK
PNP Device ID
PCI\VEN_10DF&DEV_F980&SUBSYS_F98010DF&REV_01\5&58A
F5CE&0&08
Memory Address 0xD0000000-0xDFFFFFFF
Memory Address 0xD0001000-0xD00010FF

IRQ Channel IRQ 87
Memory Address 0x0000A000-0x0000BFFF
Driver c:\windows\system32\drivers\lpnds.sys (6-5.00a11
05/14/2003 Win2003 Server 64 bits built by: WinDDK, 314.00 KB (321,536
bytes), 5/14/2003 3:44 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&385
8B390&0&08
Memory Address 0xE0200000-0xE023FFFF
Memory Address 0xE0000000-0xEFFFFFFF
I/O Port 0x0000C000-0x0000DFFF
IRQ Channel IRQ 98
Driver c:\windows\system32\drivers\hpcissb.sys (5.6.2.64 built by:
WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&1E6
C4F79&0&08
Memory Address 0xF0200000-0xF023FFFF
Memory Address 0xF0000000-0xFFFFFFFF
I/O Port 0x0000E000-0x0000FFFF
IRQ Channel IRQ 109
Driver c:\windows\system32\drivers\hpcissb.sys (5.6.2.64 built by:
WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&29A
1C50A&0&08
Memory Address 0xE0200000-0xE023FFFF
Memory Address 0xE0000000-0xEFFFFFFF
I/O Port 0x00004000-0x00005FFF
IRQ Channel IRQ 158
Driver c:\windows\system32\drivers\hpcissb.sys (5.6.2.64 built by:
WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&200
C63EC&0&08
Memory Address 0xF0200000-0xF023FFFF
Memory Address 0xF0000000-0xFFFFFFFF
I/O Port 0x00006000-0x00007FFF

Driver c:\windows\system32\drivers\hpcissb.sys (5.6.2.64 built by:
WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&37D
3486A&0&08
I/O Port 0x00008000-0x00009FFF
Memory Address 0xC0020000-0xC0020FFF
IRQ Channel IRQ 180
Driver c:\windows\system32\drivers\ql2300.sys (8.2.2.10 (W64 VI),
690.75 KB (707,328 bytes), 4/16/2003 7:45 PM)

Name Emulex LightPulse PCI Fibre Channel HBA
Manufacturer Emulex
Status OK
PNP Device ID
PCI\VEN_10DF&DEV_F980&SUBSYS_F98010DF&REV_01\5&288
74FDD&0&08
Memory Address 0xD0000000-0xDFFFFFFF
Memory Address 0xD0001000-0xD00010FF
I/O Port 0x0000A000-0x0000BFFF
IRQ Channel IRQ 191
Driver c:\windows\system32\drivers\lpnds.sys (6-5.00a11
05/14/2003 Win2003 Server 64 bits built by: WinDDK, 314.00 KB (321,536
bytes), 5/14/2003 3:44 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&225
F72B7&0&08
Memory Address 0xE0200000-0xE023FFFF
Memory Address 0xE0000000-0xEFFFFFFF
I/O Port 0x0000C000-0x0000DFFF
IRQ Channel IRQ 202
Driver c:\windows\system32\drivers\hpcissb.sys (5.6.2.64 built by:
WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&FA5
F632&0&08
Memory Address 0xF0200000-0xF023FFFF
Memory Address 0xF0000000-0xFFFFFFFF
I/O Port 0x0000E000-0x0000FFFF
IRQ Channel IRQ 213

Driver c:\windows\system32\drivers\hpcissb.sys (5.6.2.64 built by: WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

Name LSI Logic 53C1010-66 Device
Manufacturer LSI Logic Inc.
Status OK
PNP Device ID PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&194

FB0DF&0&11
I/O Port 0x00000500-0x000005FF
Memory Address 0xC001A000-0xC001A3FF
Memory Address 0xC0010000-0xC0011FFF
IRQ Channel IRQ 232
Driver c:\windows\system32\drivers\sym_u3.sys (5.2.3673.0 (_020828-1450), 66.00 KB (67,584 bytes), 10/16/2003 11:30 AM)

Name LSI Logic 53C1010-66 Device
Manufacturer LSI Logic Inc.
Status OK
PNP Device ID PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&194

FB0DF&0&11
I/O Port 0x00000600-0x000006FF
Memory Address 0xC001B000-0xC001B3FF
Memory Address 0xC0012000-0xC0013FFF
IRQ Channel IRQ 233
Driver c:\windows\system32\drivers\sym_u3.sys (5.2.3673.0 (_020828-1450), 66.00 KB (67,584 bytes), 10/16/2003 11:30 AM)

Name LSI Logic 53C1010-66 Device
Manufacturer LSI Logic Inc.
Status OK
PNP Device ID PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&194

FB0DF&0&18
I/O Port 0x00000700-0x000007FF
Memory Address 0xC001C000-0xC001C3FF
Memory Address 0xC0014000-0xC0015FFF
IRQ Channel IRQ 234
Driver c:\windows\system32\drivers\sym_u3.sys (5.2.3673.0 (_020828-1450), 66.00 KB (67,584 bytes), 10/16/2003 11:30 AM)

Name LSI Logic 53C1010-66 Device
Manufacturer LSI Logic Inc.
Status OK
PNP Device ID PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&194

FB0DF&0&19
I/O Port 0x00000800-0x000008FF
Memory Address 0xC001D000-0xC001D3FF
Memory Address 0xC0016000-0xC0017FFF
IRQ Channel IRQ 235

Driver c:\windows\system32\drivers\sym_u3.sys (5.2.3673.0 (_020828-1450), 66.00 KB (67,584 bytes), 10/16/2003 11:30 AM)
Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&548

403B&0&08
Memory Address 0xE0200000-0xE023FFFF
Memory Address 0xE0000000-0xEFFFFFFF
I/O Port 0x00004000-0x00005FFF
IRQ Channel IRQ 262
Driver c:\windows\system32\drivers\hpcissb.sys (5.6.2.64 built by: WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&ED

DA159&0&08
Memory Address 0xF0200000-0xF023FFFF
Memory Address 0xF0000000-0xFFFFFFFF
I/O Port 0x00006000-0x00007FFF
IRQ Channel IRQ 273
Driver c:\windows\system32\drivers\hpcissb.sys (5.6.2.64 built by: WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&245

142EC&0&08
I/O Port 0x00008000-0x00009FFF
Memory Address 0xC0020000-0xC0020FFF
IRQ Channel IRQ 284
Driver c:\windows\system32\drivers\ql2300.sys (8.2.2.10 (W64 V1), 690.75 KB (707,328 bytes), 4/16/2003 7:45 PM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&A46

6DE5&0&08
Memory Address 0xE0200000-0xE023FFFF
Memory Address 0xE0000000-0xEFFFFFFF
I/O Port 0x0000C000-0x0000DFFF
IRQ Channel IRQ 306
Driver c:\windows\system32\drivers\hpcissb.sys (5.6.2.64 built by: WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

Manufacturer Hewlett-Packard
Name Smart Array 5300 Controller (Non-Miniport)
Status OK
PNP Device ID PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&B10

CC7&0&08
Memory Address 0xF0200000-0xF023FFFF
Memory Address 0xF0000000-0xFFFFFFFF
I/O Port 0x0000E000-0x0000FFFF
IRQ Channel IRQ 317
Driver c:\windows\system32\drivers\hpcissb.sys (5.6.2.64 built by: WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

Name LSI Logic 53C1010-66 Device
Manufacturer LSI Logic Inc.
Status OK
PNP Device ID PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&14C

294CC&0&10
I/O Port 0x00000500-0x000005FF
Memory Address 0xC001A000-0xC001A3FF
Memory Address 0xC0010000-0xC0011FFF
IRQ Channel IRQ 336
Driver c:\windows\system32\drivers\sym_u3.sys (5.2.3673.0 (_020828-1450), 66.00 KB (67,584 bytes), 10/16/2003 11:30 AM)

Name LSI Logic 53C1010-66 Device
Manufacturer LSI Logic Inc.
Status OK
PNP Device ID PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&14C

294CC&0&11
I/O Port 0x00000600-0x000006FF
Memory Address 0xC001B000-0xC001B3FF
Memory Address 0xC0012000-0xC0013FFF
IRQ Channel IRQ 337
Driver c:\windows\system32\drivers\sym_u3.sys (5.2.3673.0 (_020828-1450), 66.00 KB (67,584 bytes), 10/16/2003 11:30 AM)

Name LSI Logic 53C1010-66 Device
Manufacturer LSI Logic Inc.
Status OK
PNP Device ID PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&14C

294CC&0&18
I/O Port 0x00000700-0x000007FF
Memory Address 0xC001C000-0xC001C3FF
Memory Address 0xC0014000-0xC0015FFF
IRQ Channel IRQ 338
Driver c:\windows\system32\drivers\sym_u3.sys (5.2.3673.0 (_020828-1450), 66.00 KB (67,584 bytes), 10/16/2003 11:30 AM)

Name LSI Logic 53C1010-66 Device
Manufacturer LSI Logic Inc.

Status OK
 PNP Device ID
 PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&14C294CC&0&19
 I/O Port 0x00000800-0x000008FF
 Memory Address 0xC001D000-0xC001D3FF
 Memory Address 0xC0016000-0xC0017FFF
 IRQ Channel IRQ 339
 Driver c:\windows\system32\drivers\sym_u3.sys (5.2.3673.0 (_020828-1450), 66.00 KB (67,584 bytes), 10/16/2003 11:30 AM)

Name Smart Array 5300 Controller (Non-Miniport)
 Manufacturer Hewlett-Packard
 Status OK
 PNP Device ID
 PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&19D51C05&0&08
 Memory Address 0xE0200000-0xE023FFFF
 Memory Address 0xE0000000-0xEFFFFFFF
 I/O Port 0x00004000-0x00005FFF
 IRQ Channel IRQ 366
 Driver c:\windows\system32\drivers\hpqcissb.sys (5.6.2.64 built by: WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

Name Smart Array 5300 Controller (Non-Miniport)
 Manufacturer Hewlett-Packard
 Status OK
 PNP Device ID
 PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&1551CFD&0&08
 Memory Address 0xF0200000-0xF023FFFF
 Memory Address 0xF0000000-0xFFFFFFFF
 I/O Port 0x00006000-0x00007FFF
 IRQ Channel IRQ 377
 Driver c:\windows\system32\drivers\hpqcissb.sys (5.6.2.64 built by: WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
 Manufacturer QLogic
 Status OK
 PNP Device ID
 PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&3666E35B&0&08
 I/O Port 0x00008000-0x00009FFF
 Memory Address 0xC0020000-0xC0020FFF
 IRQ Channel IRQ 388
 Driver c:\windows\system32\drivers\ql2300.sys (8.2.2.10 (W64 VI), 690.75 KB (707,328 bytes), 4/16/2003 7:45 PM)

Name Smart Array 5300 Controller (Non-Miniport)
 Manufacturer Hewlett-Packard
 Status OK

PNP Device ID
 PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&23C36708&0&08
 Memory Address 0xE0200000-0xE023FFFF
 Memory Address 0xE0000000-0xEFFFFFFF
 I/O Port 0x0000C000-0x0000DFFF
 IRQ Channel IRQ 410
 Driver c:\windows\system32\drivers\hpqcissb.sys (5.6.2.64 built by: WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

Name Smart Array 5300 Controller (Non-Miniport)
 Manufacturer Hewlett-Packard
 Status OK
 PNP Device ID
 PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&2D6138E4&0&08
 Memory Address 0xF0200000-0xF023FFFF
 Memory Address 0xF0000000-0xFFFFFFFF
 I/O Port 0x0000E000-0x0000FFFF
 IRQ Channel IRQ 421
 Driver c:\windows\system32\drivers\hpqcissb.sys (5.6.2.64 built by: WinDDK, 101.50 KB (103,936 bytes), 9/29/2003 10:02 AM)

[IDE]

Item	Value
[Printing]	

Name	Driver	Port Name	Server Name
[Problem Devices]			
Device	PNP Device ID	Error Code	
Not Available ACPI\HWP5003\2&DABA3FF&0			The drivers for this device are not installed.

[USB]

Device	PNP Device ID
[Software Environment]	

[System Drivers]

Name	Description	File	Type	Started	Start Mode
Stop	State	Status	Error Control	Accept	Pause
4mmdat	4mmdat	c:\windows\system32\drivers\4mmdat.sys	Kernel Driver	No	Manual
	Normal	No	No	Stopped	OK

Device	Driver	Status	Kernel Driver	Normal	Kernel Driver
abiosdsk	Abiosdsk	Not Available	Kernel Driver	No	Disabled
acpi	Microsoft ACPI Driver	Stopped	OK	No	No
	c:\windows\system32\drivers\acpi.sys	Kernel Driver	Yes		
	Boot	Running	OK	Normal	No
	Yes				
acpiec	ACPIEC	c:\windows\system32\drivers\acpiec.sys	Kernel Driver	No	Normal
	No	Disabled	Stopped	OK	
	No	No			
adpu160m	adpu160m	c:\windows\system32\drivers\adpu160m.sys	Kernel Driver	No	OK
	Normal	No	Disabled	Stopped	
	No	No			
adpu320	adpu320	c:\windows\system32\drivers\adpu320.sys	Kernel Driver	No	OK
	Normal	No	Disabled	Stopped	
	No	No			
afcnt	afcnt	c:\windows\system32\drivers\afcnt.sys	Kernel Driver	No	Normal
	No	Disabled	Stopped	OK	
	No	No			
afd	AFD Networking Support Environment	c:\windows\system32\drivers\afd.sys	Kernel Driver	Yes	
	Auto	Running	OK	Normal	No
	Yes				
agp460	Intel AGP Bus Filter	c:\windows\system32\drivers\agp460.sys	Kernel Driver	No	Normal
	No	Disabled	Stopped	OK	
	No	No			
aic78u2	aic78u2	c:\windows\system32\drivers\aic78u2.sys	Kernel Driver	No	OK
	Normal	No	Disabled	Stopped	
	No	No			
aic78xx	aic78xx	c:\windows\system32\drivers\aic78xx.sys	Kernel Driver	No	OK
	Normal	No	Disabled	Stopped	
	No	No			
aliide	AliIde	c:\windows\system32\drivers\aliide.sys	Kernel Driver	No	Normal
	No	Disabled	Stopped	OK	
	No	No			
asynmac	RAS Asynchronous Media Driver	c:\windows\system32\drivers\asynmac.sys	Kernel Driver	No	Normal
	No	Manual	Stopped	OK	
	No	No			
atapi	Standard IDE/ESDI Hard Disk Controller	c:\windows\system32\drivers\atapi.sys	Kernel Driver	No	No
	Disabled	Stopped	OK	Normal	
	No				
atdisk	Atdisk	Not Available	Kernel Driver	No	Disabled
	Stopped	OK	Ignore	No	No
atmarpc	ATM ARP Client Protocol	c:\windows\system32\drivers\atmarpc.sys	Kernel Driver	No	Normal
	No	Manual	Stopped	OK	
	No	No			

audstub	Audio Stub Driver	c:\windows\system32\drivers\audstub.sys	Kernel		
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
b57nd	Broadcom NetXtreme Gigabit Ethernet	c:\windows\system32\drivers\b57xp64.sys	Kernel		
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
beep	Beep	c:\windows\system32\drivers\beep.sys	Kernel		
Driver	Yes	System	Running	OK	Normal
	No	Yes			
cbidf	cbidf	c:\windows\system32\drivers\cbidf2k.sys	Kernel Driver	No	Disabled Stopped OK
	Normal	No	No		
cbidf2k	cbidf2k	c:\windows\system32\drivers\cbidf2k.sys	Kernel Driver	No	Disabled Stopped OK
	Normal	No	No		
cdfs	Cdfs	c:\windows\system32\drivers\cdfs.sys	File		
System Driver	Yes	Disabled	Running	OK	
	Normal	No	Yes		
cdrom	CD-ROM Driver	c:\windows\system32\drivers\cdrom.sys	Kernel Driver	Yes	
	System	Running	OK	Normal	No
	Yes				
changer	Changer	Not Available	Kernel Driver	No	System
	Stopped	OK	Ignore	No	No
clusdisk	Cluster Disk Driver	c:\windows\system32\drivers\clusdisk.sys	Kernel		
Driver	No	Disabled	Stopped	OK	Normal
	No	No			
cmdide	CmdIde	c:\windows\system32\drivers\cmdide.sys	Kernel Driver	No	Disabled Stopped OK
	Normal	No	No		
cpqarry2	cpqarry2	c:\windows\system32\drivers\cpqarry2.sys	Kernel Driver	No	Disabled Stopped OK
	Normal	No	No		
cpqcisse	CPQCISSE	c:\windows\system32\drivers\cpqcisse.sys	Kernel Driver	No	Manual Stopped OK
	Normal	No	No		
cpqcissm	cpqcissm	c:\windows\system32\drivers\cpqcissm.sys	Kernel Driver	Yes	Boot Running OK
	Normal	No	Yes		
cpqfcalm	cpqfcalm	c:\windows\system32\drivers\cpqfcalm.sys	Kernel Driver	No	Disabled Stopped OK
	Normal	No	No		
crcdisk	CRC Disk Filter Driver	c:\windows\system32\drivers\crcdisk.sys	Kernel		
Driver	Yes	Boot	Running	OK	Normal
	No	Yes			

dfsdriver	DfsDriver	c:\windows\system32\drivers\dfs.sys	File		
System Driver	Yes	Boot	Running	OK	
disk	disk	c:\windows\system32\drivers\disk.sys	Kernel		
Driver	Yes	Boot	Running	OK	Normal
	No	Yes			
dmbboot	dmbboot	c:\windows\system32\drivers\dmbboot.sys	Kernel Driver	Yes	Disabled Running OK
	Normal	No	Yes		
dmio	Logical Disk Manager Driver	c:\windows\system32\drivers\dmio.sys	Kernel Driver	Yes	
	Boot	Running	OK	Normal	No
	Yes				
dmload	dmload	c:\windows\system32\drivers\dmload.sys	Kernel Driver	Yes	Boot Running OK
	Normal	No	Yes		
dpti2o	dpti2o	c:\windows\system32\drivers\dpti2o.sys	Kernel		
Driver	No	Disabled	Stopped	OK	Normal
	No	No			
e1000	Intel(R) PRO/1000 Device Driver	c:\windows\system32\drivers\e1000645.sys	Kernel		
Driver	No	Manual	Stopped	OK	Normal
	No	No			
fastfat	Fastfat	c:\windows\system32\drivers\fastfat.sys	File		
System Driver	Yes	Disabled	Running	OK	
	Normal	No	Yes		
fdc	Fdc	c:\windows\system32\drivers\fdc.sys	Kernel		
Driver	No	System	Stopped	OK	Ignore
	No	No			
filemon	FILEMON	\\?\c:\windows\system32\drivers\filemon.sys	Kernel Driver	No	Manual Stopped OK
	Normal	No	No		
fips	Fips	c:\windows\system32\drivers\fips.sys	Kernel		
Driver	Yes	System	Running	OK	Normal
	No	Yes			
flpydisk	Flpydisk	c:\windows\system32\drivers\flpydisk.sys	Kernel Driver	No	System Stopped OK
	Ignore	No	No		
ftdisk	Volume Manager Driver	c:\windows\system32\drivers\ftdisk.sys	Kernel Driver	Yes	
	Boot	Running	OK	Normal	No
	Yes				
gpc	Generic Packet Classifier	c:\windows\system32\drivers\msgpc.sys	Kernel Driver	Yes	
	Manual	Running	OK	Normal	No
	Yes				
hidusb	Microsoft HID Class Driver	c:\windows\system32\drivers\hidusb.sys	Kernel Driver	No	
	Manual	Stopped	OK	Ignore	No
	No				
hphlth	HP Baseboard Management Controller Interface Driver	c:\windows\system32\drivers\hphlth.sys	Kernel Driver	Yes	
	Manual	Running	OK	Normal	No
	Yes				

hmpmpser	HP MP Driver	c:\windows\system32\drivers\hmpmpser.sys	Kernel Driver	Yes	Manual Running OK
	Normal	No	Yes		
hpn	hpn	c:\windows\system32\drivers\hpn.sys	Kernel		
Driver	No	Disabled	Stopped	OK	Normal
	No	No			
hpqcissb	Smart Array Controllers Non-Miniport Bus Driver	c:\windows\system32\drivers\hpqcissb.sys	Kernel		
Driver	Yes	Boot	Running	OK	Normal
	No	Yes			
hpqcissd	Smart Array Controllers Non-Miniport Disk Driver	c:\windows\system32\drivers\hpqcissd.sys	Kernel		
Driver	Yes	Boot	Running	OK	Normal
	No	Yes			
http	HTTP	c:\windows\system32\drivers\http.sys	Kernel		
Driver	No	Manual	Stopped	OK	Normal
	No	No			
i2omgmt	i2omgmt	Not Available	Kernel Driver	No	System
	Stopped	OK	Normal	No	No
imapi	CD-Burning Filter Driver	c:\windows\system32\drivers\imapi.sys	Kernel Driver	No	
	System	Stopped	OK	Normal	No
	No				
intelide	IntelIde	c:\windows\system32\drivers\intelide.sys	Kernel Driver	No	Disabled Stopped OK
	Normal	No	No		
ipfilterdriver	IP Traffic Filter Driver	c:\windows\system32\drivers\ipfltdrv.sys	Kernel		
Driver	No	Manual	Stopped	OK	Normal
	No	No			
ipinip	IP in IP Tunnel Driver	c:\windows\system32\drivers\ipinip.sys	Kernel Driver	No	
	Manual	Stopped	OK	Normal	No
	No				
ipnat	IP Network Address Translator	c:\windows\system32\drivers\ipnat.sys	Kernel Driver	No	
	Manual	Stopped	OK	Normal	No
	No				
ipsec	IPSEC driver	c:\windows\system32\drivers\ipsec.sys	Kernel		
Driver	Yes	System	Running	OK	Normal
	No	Yes			
isapnp	PnP ISA/EISA Bus Driver	c:\windows\system32\drivers\isapnp.sys	Kernel Driver	No	
	Disabled	Stopped	OK	Critical	No
	No				
kbdclass	Keyboard Class Driver	c:\windows\system32\drivers\kbdclass.sys	Kernel		
Driver	Yes	System	Running	OK	Normal
	No	Yes			
kbdhid	Keyboard HID Driver	c:\windows\system32\drivers\kbdhid.sys	Kernel Driver	No	
	System	Stopped	OK	Ignore	No
	No				

ksecdd	KSecDD	c:\windows\system32\drivers\ksecdd.sys	Kernel Driver	Yes	Boot	Running	OK
	Normal	No	Yes				
ksthunk	Kernel Streaming WOW64 Thunk Service	c:\windows\system32\drivers\ksthunk.sys	Kernel Driver	No	Manual	Stopped	OK
	Normal	No	No				
lp6nds35	Ip6nds35	c:\windows\system32\drivers\lp6nds35.sys	Kernel Driver	No	Disabled	Stopped	OK
	Normal	No	No				
lpxftr	lpxftr	c:\windows\system32\drivers\lpxftr.sys	Kernel Driver	Yes	Boot	Running	OK
	Normal	No	Yes				
lpxnds	lpxnds	c:\windows\system32\drivers\lpxnds.sys	Kernel Driver	Yes	Boot	Running	OK
	Normal	No	Yes				
mnmd	mnmd	Not Available	Kernel Driver	No	Ignore	No	System
	Stopped	OK	Ignore	No			No
modem	Modem	c:\windows\system32\drivers\modem.sys	Kernel Driver	No	Manual	Stopped	OK
	Ignore	No	No				
mouclass	Mouse Class Driver	c:\windows\system32\drivers\mouclass.sys	Kernel Driver	Yes	System	Running	OK
	Normal	No	Yes				
mouhid	Mouse HID Driver	c:\windows\system32\drivers\mouhid.sys	Kernel Driver	No	Manual	Stopped	OK
	Normal	No	No				
mountmgr	Mount Point Manager	c:\windows\system32\drivers\mountmgr.sys	Kernel Driver	Yes	Boot	Running	OK
	Normal	No	Yes				
mraid35x	mraid35x	c:\windows\system32\drivers\mraid35x.sys	Kernel Driver	No	Disabled	Stopped	OK
	Normal	No	No				
mrxdav	WebDav Client Redirector	c:\windows\system32\drivers\mrxdav.sys	File System Driver	No	Manual	Stopped	OK
	Normal	No	No				
mrxsmb	MRXSMB	c:\windows\system32\drivers\mrxsmb.sys	File System Driver	Yes	System	Running	OK
	OK	Normal	No				
msfs	Msfs	c:\windows\system32\drivers\msfs.sys	File System Driver	Yes	System	Running	OK
	Normal	No	Yes				
mup	Mup	c:\windows\system32\drivers\mup.sys	File System Driver	Yes	Boot	Running	OK
	Normal	No	Yes				

ndis	NDIS System Driver	c:\windows\system32\drivers\ndis.sys	Kernel Driver	Yes	Boot	Running	OK
	Normal	No	Yes				
ndistapi	Remote Access NDIS TAPI Driver	c:\windows\system32\drivers\ndistapi.sys	Kernel Driver	Yes	Manual	Running	OK
	Normal	No	Yes				
ndisuio	NDIS Usermode I/O Protocol	c:\windows\system32\drivers\ndisuio.sys	Kernel Driver	No	Manual	Stopped	OK
	Normal	No	No				
ndiswan	Remote Access NDIS WAN Driver	c:\windows\system32\drivers\ndiswan.sys	Kernel Driver	Yes	Manual	Running	OK
	Normal	No	Yes				
ndproxy	NDIS Proxy	c:\windows\system32\drivers\ndproxy.sys	Kernel Driver	Yes	Manual	Running	OK
	Normal	No	Yes				
netbios	NetBIOS Interface	c:\windows\system32\drivers\netbios.sys	File System Driver	Yes	System	Running	OK
	Normal	No	Yes				
netbt	NetBios over Tcpip	c:\windows\system32\drivers\netbt.sys	Kernel Driver	Yes	System	Running	OK
	Normal	No	Yes				
nfrd960	nfrd960	c:\windows\system32\drivers\nfrd960.sys	Kernel Driver	No	Disabled	Stopped	OK
	Normal	No	No				
npfs	Npfs	c:\windows\system32\drivers\npfs.sys	File System Driver	Yes	System	Running	OK
	Normal	No	Yes				
ntfs	Ntfs	c:\windows\system32\drivers\ntfs.sys	File System Driver	Yes	Disabled	Running	OK
	Normal	No	Yes				
null	Null	c:\windows\system32\drivers\null.sys	Kernel Driver	Yes	System	Running	OK
	Normal	No	Yes				
partmgr	Partition Manager	c:\windows\system32\drivers\partmgr.sys	Kernel Driver	Yes	Boot	Running	OK
	Normal	No	Yes				
pci	PCI Bus Driver	c:\windows\system32\drivers\pci.sys	Kernel Driver	Yes	Boot	Running	OK
	Critical	No	Yes				
pciide	PCIIde	c:\windows\system32\drivers\pciide.sys	Kernel Driver	No	Disabled	Stopped	OK
	Normal	No	No				
pcmcia	Pcmcia	c:\windows\system32\drivers\pcmcia.sys	Kernel Driver	No	Disabled	Stopped	OK
	Normal	No	No				

pdcomp	PDCOMP	Not Available	Kernel Driver	No	Ignore	No	Manual
	Stopped	OK	Ignore	No			No
pdframe	PDFRAME	Not Available	Kernel Driver	No	Ignore	No	Manual
	Stopped	OK	Ignore	No			No
pdreli	PDRELI	Not Available	Kernel Driver	No	Ignore	No	Manual
	Stopped	OK	Ignore	No			No
pdrframe	PDRFRAME	Not Available	Kernel Driver	No	Ignore	No	Manual
	Stopped	OK	Ignore	No			No
pptpminiport	WAN Miniport (PPTP)	c:\windows\system32\drivers\rasppptp.sys	Kernel Driver	Yes	Manual	Running	OK
	Normal	No	Yes				
processor	Processor Driver	c:\windows\system32\drivers\processr.sys	Kernel Driver	Yes	Manual	Running	OK
	Normal	No	Yes				
ptilink	Direct Parallel Link Driver	c:\windows\system32\drivers\ptilink.sys	Kernel Driver	Yes	Manual	Running	OK
	Normal	No	Yes				
ql1080	ql1080	c:\windows\system32\drivers\ql1080.sys	Kernel Driver	No	Disabled	Stopped	OK
	Normal	No	No				
ql10wnt	Ql10wnt	Not Available	Kernel Driver	No	Normal	No	Disabled
	Stopped	OK	Normal	No			No
ql12160	ql12160	c:\windows\system32\drivers\ql12160.sys	Kernel Driver	No	Disabled	Stopped	OK
	Normal	No	No				
ql1240	ql1240	c:\windows\system32\drivers\ql1240.sys	Kernel Driver	No	Disabled	Stopped	OK
	Normal	No	No				
ql1280	ql1280	c:\windows\system32\drivers\ql1280.sys	Kernel Driver	No	Disabled	Stopped	OK
	Normal	No	No				
ql2100	ql2100	c:\windows\system32\drivers\ql2100.sys	Kernel Driver	No	Disabled	Stopped	OK
	Normal	No	No				
ql2200	ql2200	c:\windows\system32\drivers\ql2200.sys	Kernel Driver	No	Disabled	Stopped	OK
	Normal	No	No				
ql2300	ql2300	c:\windows\system32\drivers\ql2300.sys	Kernel Driver	Yes	Boot	Running	OK
	Normal	Yes	Yes				
qlvika	qlvika	c:\windows\system32\drivers\qlvika.sys	Kernel Driver	Yes	Auto	Running	OK
	Normal	No	Yes				

rasacd	Remote Access Auto Connection Driver	c:\windows\system32\drivers\rasacd.sys	Kernel Driver	Yes	
	System	Running	OK	Normal	No
rasl2tp	WAN Miniport (L2TP)	c:\windows\system32\drivers\rasl2tp.sys	Kernel Driver	Yes	
	Manual	Running	OK	Normal	No
rasppoe	Remote Access PPPOE Driver	c:\windows\system32\drivers\rasppoe.sys			Kernel
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
raspti	Direct Parallel	c:\windows\system32\drivers\raspti.sys	Kernel Driver	Yes	
	Manual	Running	OK	Normal	No
rdbss	Rdbss	c:\windows\system32\drivers\rdbss.sys	File		
System Driver	Normal	No	Yes	System	Running
	Yes			OK	
rdpcdd	RDPCDD	c:\windows\system32\drivers\rdpcdd.sys			
	Kernel Driver	Yes	System	Running	OK
	Ignore	No	Yes		
rdpdr	Terminal Server Device Redirector Driver	c:\windows\system32\drivers\rdpdr.sys	Kernel Driver	Yes	
	Manual	Running	OK	Normal	No
	Yes				
rdpwd	RDPWD	c:\windows\system32\drivers\rdpwd.sys	Kernel		
Driver	Yes	Manual	Running	OK	Ignore
	No	Yes			
redbook	Digital CD Audio Playback Filter Driver	c:\windows\system32\drivers\redbook.sys			Kernel
Driver	Yes	System	Running	OK	Normal
	No	Yes			
serenum	Serenum Filter Driver	c:\windows\system32\drivers\serenum.sys			Kernel
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
serial	Serial	c:\windows\system32\drivers\serial.sys	Kernel		
Driver	No	Auto	Stopped	OK	Ignore
	No	No			
sfloppy	Sfloppy	c:\windows\system32\drivers\sfloppy.sys			
	Kernel Driver	No	System	Stopped	OK
	Ignore	No	No		
simbad	Simbad	Not Available	Kernel Driver	No	Disabled
	Stopped	OK	Normal	No	No
srv	Srv	c:\windows\system32\drivers\srv.sys	File		
System Driver	Normal	Yes	Manual	Running	OK
	No	Yes			

swenum	Software Bus Driver	c:\windows\system32\drivers\swenum.sys	Kernel		
Driver	Yes	Manual	Running	OK	Normal
symc8xx	symc8xx	c:\windows\system32\drivers\symc8xx.sys			
	Kernel Driver	No	Disabled	Stopped	OK
	Normal	No	No		
symmpi	symmpi	c:\windows\system32\drivers\symmpi.sys			
	Kernel Driver	No	Disabled	Stopped	OK
	Normal	No	No		
sym_hi	sym_hi	c:\windows\system32\drivers\sym_hi.sys			
	Kernel Driver	No	Disabled	Stopped	OK
	Normal	No	No		
sym_u3	sym_u3	c:\windows\system32\drivers\sym_u3.sys			
	Kernel Driver	Yes	Boot	Running	OK
	Normal	No	Yes		
tcpip	TCP/IP Protocol Driver	c:\windows\system32\drivers\tcpip.sys	Kernel Driver	Yes	
	System	Running	OK	Normal	No
	Yes				
tdpipe	TDPIPE	c:\windows\system32\drivers\tdpipe.sys	Kernel		
Driver	No	Manual	Stopped	OK	Ignore
	No	No			
tdtcp	TDTCP	c:\windows\system32\drivers\tdtcp.sys	Kernel		
Driver	Yes	Manual	Running	OK	Ignore
	No	Yes			
termdd	Terminal Device Driver	c:\windows\system32\drivers\termdd.sys	Kernel		
Driver	Yes	System	Running	OK	Normal
	No	Yes			
toside	TosIde	c:\windows\system32\drivers\toside.sys	Kernel		
Driver	No	Disabled	Stopped	OK	Normal
	No	No			
udfs	Udfs	c:\windows\system32\drivers\udfs.sys	File		
System Driver	No	Disabled	Stopped	OK	
	Normal	No	No		
usbccgp	Microsoft USB Generic Parent Driver	c:\windows\system32\drivers\usbccgp.sys	Kernel		
Driver	No	Manual	Stopped	OK	Normal
	No	No			
usbehci	Microsoft USB 2.0 Enhanced Host Controller Miniport Driver	c:\windows\system32\drivers\usbehci.sys	Kernel		
Driver	No	Manual	Stopped	OK	Normal
	No	No			
usbhub	USB2 Enabled Hub	c:\windows\system32\drivers\usbhub.sys	Kernel		
Driver	No	Manual	Stopped	OK	Normal
	No	No			
usbohci	Microsoft USB Open Host Controller Miniport Driver	c:\windows\system32\drivers\usbohci.sys	Kernel		
Driver	No	Manual	Stopped	OK	Normal
	No	No			

vga	vga	c:\windows\system32\drivers\vgapnp.sys			
	Kernel Driver	No	Manual	Stopped	OK
vgasave	VGA Display Controller	c:\windows\system32\drivers\vgasave.sys	Kernel Driver	No	
	System	Stopped	OK	Ignore	No
	No				
viaide	ViaIde	c:\windows\system32\drivers\viaide.sys	Kernel		
Driver	No	Disabled	Stopped	OK	Normal
	No	No			
volsnap	Storage volumes	c:\windows\system32\drivers\volsnap.sys			Kernel
Driver	Yes	Boot	Running	OK	Normal
	No	Yes			
wanarp	Remote Access IP ARP Driver	c:\windows\system32\drivers\wanarp.sys			Kernel
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
wdica	WDICA	Not Available	Kernel Driver	No	Manual
	Stopped	OK	Ignore	No	No
wlbs	Network Load Balancing	c:\windows\system32\drivers\wlbs.sys	Kernel Driver	No	
	Manual	Stopped	OK	Normal	No
	No				

[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	Not Available	Not Available
ACPI IA64-based PC	Yes	COMPUTER	5.2.3790.0	hal.inf
Available	10/1/2002	(Standard computers)	hal.inf	Not
Available	ROOT\ACPI_HAL\0000			
Microsoft ACPI-Compliant System	Yes	SYSTEM	5.2.3790.0	10/1/2002
Available	ACPI_HAL\PNP0C08\0	Microsoft	acpi.inf	Not
ACPI Power Button	Yes	SYSTEM	5.2.3790.0	10/1/2002
Available	ACPI\PNP0C0C\2&DABA3FF&0	(Standard system devices)	machine.inf	Not
Not Available	Not Available	Not Available	Not Available	Not Available
Not Available	Not Available	Not Available	Not Available	Not Available
HP System Memory Device	Yes	SYSTEM	5.0.0.2	6/19/2003
	HP	oem5.inf	Not Available	
	ACPI\HWP1000\A56E04371033			
HP System Memory Device	Yes	SYSTEM	5.0.0.2	6/19/2003
	HP	oem5.inf	Not Available	
	ACPI\HWP1000\A56E04371034			
HP System Memory Device	Yes	SYSTEM	5.0.0.2	6/19/2003
	HP	oem5.inf	Not Available	
	ACPI\HWP1000\A56E0439686			

HP System Memory Device	Yes	SYSTEM	5.0.0.2	
6/19/2003 HP	oem5.inf	Not Available		
ACPI\HWP1000\A56E04399827				
HP System Memory Device	Yes	SYSTEM	5.0.0.2	
6/19/2003 HP	oem5.inf	Not Available		
ACPI\HWP1000\A56E04399888				
HP System Memory Device	Yes	SYSTEM	5.0.0.2	
6/19/2003 HP	oem5.inf	Not Available		
ACPI\HWP1000\A56E04410051				
HP System Memory Device	Yes	SYSTEM	5.0.0.2	
6/19/2003 HP	oem5.inf	Not Available		
ACPI\HWP1000\A56E04409964				
HP System Memory Device	Yes	SYSTEM	5.0.0.2	
6/19/2003 HP	oem5.inf	Not Available		
ACPI\HWP1000\A56E04409899				
HP System Memory Device	Yes	SYSTEM	5.0.0.2	
6/19/2003 HP	oem5.inf	Not Available		
ACPI\HWP1000\A56E04399862				
HP System Memory Device	Yes	SYSTEM	5.0.0.2	
6/19/2003 HP	oem5.inf	Not Available		
ACPI\HWP1000\A56E04399859				
HP System Memory Device	Yes	SYSTEM	5.0.0.2	
6/19/2003 HP	oem5.inf	Not Available		
ACPI\HWP1000\A56E04399796				
HP System Memory Device	Yes	SYSTEM	5.0.0.2	
6/19/2003 HP	oem5.inf	Not Available		
ACPI\HWP1000\A56E04399967				
HP System Memory Device	Yes	SYSTEM	5.0.0.2	
6/19/2003 HP	oem5.inf	Not Available		
ACPI\HWP1000\A56E04399778				
HP System Memory Device	Yes	SYSTEM	5.0.0.2	
6/19/2003 HP	oem5.inf	Not Available		
ACPI\HWP1000\A56E04409963				
HP System Memory Device	Yes	SYSTEM	5.0.0.2	
6/19/2003 HP	oem5.inf	Not Available		
ACPI\HWP1000\A56E04409961				
HP System Memory Device	Yes	SYSTEM	5.0.0.2	
6/19/2003 HP	oem5.inf	Not Available		
ACPI\HWP1000\A56E04410178				
Processor	Yes	PROCESSOR	5.2.3790.0	10/1/2002 (Standard processor types)
	cpu.inf	Not Available	ACPI\GENUINEINTEL_-IA64_FAMILY_31_MODEL_1\0	
Processor	Yes	PROCESSOR	5.2.3790.0	10/1/2002 (Standard processor types)
	cpu.inf	Not Available	ACPI\GENUINEINTEL_-IA64_FAMILY_31_MODEL_1\1	
Processor	Yes	PROCESSOR	5.2.3790.0	10/1/2002 (Standard processor types)
	cpu.inf	Not Available	ACPI\GENUINEINTEL_-IA64_FAMILY_31_MODEL_1\2	
Processor	Yes	PROCESSOR	5.2.3790.0	10/1/2002 (Standard processor types)
	cpu.inf	Not Available	ACPI\GENUINEINTEL_-IA64_FAMILY_31_MODEL_1\3	

Generic Bus	Yes	SYSTEM	5.2.3790.0	10/1/2002 (Standard system devices)
	machine.inf	Not Available	ACPI\HWP0005\1	
Generic Bus	Yes	SYSTEM	5.2.3790.0	10/1/2002 (Standard system devices)
	machine.inf	Not Available	ACPI\HWP0004\100	
PCI bus	Yes	SYSTEM	5.2.3790.0	10/1/2002 (Standard system devices)
	machine.inf	Not Available	ACPI\HWP0002\20	
HP MP Serial AUX/UPS Port	Yes	PORTS	5.0.3663.16	
12/12/2002 Hewlett Packard Co.	oem1.inf	Not Available		
Available PCI\VEN_103C&DEV_1290&SUBSYS_1291103C&REV_01\5&5CDAB61&0&00				
HP Management Processor	Yes	SYSTEM	5.0.3790.0	
7/10/2003 Hewlett-Packard Co	oem0.inf	Not Available		
Available PCI\VEN_103C&DEV_1048&SUBSYS_1282103C&REV_03\5&5CDAB61&0&01				
Broadcom NetXtreme Gigabit Ethernet	Yes	NET	7.15.0.0	
8/29/2003 Broadcom	oem2.inf	Not Available		
PCI\VEN_14E4&DEV_1645&SUBSYS_1300103C&REV_15\5&5CDAB61&0&08				
LSI Logic 53C1010-66 Device	Yes	SCSIADAPTER		
5.2.3790.0 10/1/2002 LSI Logic Inc.				
pnpscsi.inf	Not Available			
PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&5CDAB61&0&10				
Disk drive	Yes	DISKDRIVE	5.2.3790.0	10/1/2002 (Standard disk drives)
	disk.inf	Not Available		
SCSI\DISK&VEN_HP_18.2G&PROD_ST318452LC&REV_HP02\6&2ACB239A&0&060				
LSI Logic 53C1010-66 Device	Yes	SCSIADAPTER		
5.2.3790.0 10/1/2002 LSI Logic Inc.				
pnpscsi.inf	Not Available			
PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&5CDAB61&0&11				
CD-ROM Drive	Yes	CDROM	5.2.3790.0	
10/1/2002 (Standard CD-ROM drives)	cdrom.inf	Not Available		
Available SCSI\CDROM&VEN_HP&PROD_DVD-ROM_305&REV_1.01\6&181172F5&0&020				
LSI Logic 53C1010-66 Device	Yes	SCSIADAPTER		
5.2.3790.0 10/1/2002 LSI Logic Inc.				
pnpscsi.inf	Not Available			
PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&5CDAB61&0&18				
LSI Logic 53C1010-66 Device	Yes	SCSIADAPTER		
5.2.3790.0 10/1/2002 LSI Logic Inc.				
pnpscsi.inf	Not Available			
PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&5CDAB61&0&19				
HP Baseboard Management Controller Interface Driver	Yes			
SYSTEM	6.30.3790.6	6/17/2003 Hewlett Packard Co.		
oem8.inf	Not Available	ACPI\IPI0001\1		

PCI bus	Yes	SYSTEM	5.2.3790.0	10/1/2002 (Standard system devices)
	machine.inf	Not Available	ACPI\HWP0002\22	
PCI bus	Yes	SYSTEM	5.2.3790.0	10/1/2002 (Standard system devices)
	machine.inf	Not Available	ACPI\HWP0002\24	
PCI bus	Yes	SYSTEM	5.2.3790.0	10/1/2002 (Standard system devices)
	machine.inf	Not Available	ACPI\HWP0002\28	
Smart Array 5300 Controller (Non-Miniport)	No			
SCSIADAPTER	5.6.59.64	4/8/2003 Hewlett-Packard		
oem14.inf	Not Available			
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&28CA0570&0&08				
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64	
4/8/2003 Hewlett-Packard	oem15.inf	Not Available		
Available HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&105C4AD1&0&0000000400000000				
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64	
4/8/2003 Hewlett-Packard	oem15.inf	Not Available		
Available HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&105C4AD1&0&0100004000000000				
PCI bus	Yes	SYSTEM	5.2.3790.0	10/1/2002 (Standard system devices)
	machine.inf	Not Available	ACPI\HWP0002\2C	
Smart Array 5300 Controller (Non-Miniport)	No			
SCSIADAPTER	5.6.59.64	4/8/2003 Hewlett-Packard		
oem14.inf	Not Available			
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&1F34A452&0&08				
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64	
4/8/2003 Hewlett-Packard	oem15.inf	Not Available		
Available HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&59190E1&0&0000000400000000				
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64	
4/8/2003 Hewlett-Packard	oem15.inf	Not Available		
Available HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&59190E1&0&0100004000000000				
Generic Bus	Yes	SYSTEM	5.2.3790.0	10/1/2002 (Standard system devices)
	machine.inf	Not Available	ACPI\HWP0004\101	
PCI bus	Yes	SYSTEM	5.2.3790.0	10/1/2002 (Standard system devices)
	machine.inf	Not Available	ACPI\HWP0002\30	
QLogic QLA23xx PCI Fibre Channel Adapter	Yes			
SCSIADAPTER	8.2.2.10	4/16/2003 QLogic		
oem9.inf	Not Available			
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&9C102BF&0&08				

Qlogic processor device Yes SYSTEM 5.2.3790.0
10/1/2002 QLOGIC scsdev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_6&23AE35BF&0&07F0
PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\34

Emulex LightPulse PCI Fibre Channel HBA Yes
SCSIADAPTER 6.5.0.11 5/14/2003 Emulex
oem4.inf Not Available
PCI\VEN_10DF&DEV_F980&SUBSYS_F98010DF&REV_01\5&58A
F5CE&0&08

Compaq MSA1000 Yes SYSTEM 5.2.3790.0
10/1/2002 Compaq scsdev.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\6&13
0CB135&0&17C0
Disk drive Yes DISKDRIVE 5.2.3790.0 10/1/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3
8\6&130CB135&0&17C1
Disk drive Yes DISKDRIVE 5.2.3790.0 10/1/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.3
8\6&130CB135&0&17C2
Emulex Simulate Device Yes SYSTEM 5.1.0.0
2/26/2003 Emulex oem7.inf Not Available
SCSI\ASCIT8&VEN_EMULEX&PROD_UTILITY_IFC_DEV.&REV_E
MU2\6&130CB135&0&27F0
PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\38

Smart Array 5300 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem14.inf Not Available
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&385
8B390&0&08

Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem15.inf Not

Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&12
5643B8&0&0000004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem15.inf Not

Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&12
5643B8&0&0100004000000000
PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\3C

Smart Array 5300 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem14.inf Not Available
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&1E6
5643B8&0&0000004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem15.inf Not

Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&3A
74778&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem15.inf Not

Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&3A
74778&0&0100004000000000
ACPI Fixed Feature Button Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices) machine.inf Not

Available ACPI\FIXEDBUTTON\2&DABA3FF&0
Processor Yes PROCESSOR 5.2.3790.0 10/1/2002 (Standard
processor types) cpu.inf Not Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1_4
Processor Yes PROCESSOR 5.2.3790.0 10/1/2002 (Standard
processor types) cpu.inf Not Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1_5
Processor Yes PROCESSOR 5.2.3790.0 10/1/2002 (Standard
processor types) cpu.inf Not Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1_6
Processor Yes PROCESSOR 5.2.3790.0 10/1/2002 (Standard
processor types) cpu.inf Not Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1_7
Generic Bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0005\0

Generic Bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0004\0

PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\0

PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\2

PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\4

PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\8

Smart Array 5300 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem14.inf Not Available
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&29A
1C50A&0&08

Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem15.inf Not

Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&18
8E934&0&0000000000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem15.inf Not

Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&18
8E934&0&0100004000000000
PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\C

Smart Array 5300 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem14.inf Not Available
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&200
C63EC&0&08

Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem15.inf Not

Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&2C
BB6481&0&0000004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem15.inf Not

Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&2C
BB6481&0&0100004000000000
Generic Bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0004\1

PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\10

QLogic QLA23xx PCI Fibre Channel Adapter Yes
SCSIADAPTER 8.2.2.10 4/16/2003 QLogic
oem9.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&37D
3486A&0&08

Qlogic processor device Yes SYSTEM 5.2.3790.0
10/1/2002 QLOGIC scsdev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_6&2DDF6751&0&07F0
PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\HWP0002\14

Emulex LightPulse PCI Fibre Channel HBA Yes
SCSIADAPTER 6.5.0.11 5/14/2003 Emulex
oem4.inf Not Available
PCI\VEN_10DF&DEV_F980&SUBSYS_F98010DF&REV_01\5&288
74FDD&0&08

Compaq MSA1000	Yes	SYSTEM	5.2.3790.0		
10/1/2002	Compaq	scsidev.inf	Not Available		
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_2.38\6&37B414EC&0&17C0					
Disk drive	Yes	DISKDRIVE	5.2.3790.0	10/1/2002	(Standard disk drives)
disk.inf	Not Available				
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\6&37B414EC&0&17C1					
Disk drive	Yes	DISKDRIVE	5.2.3790.0	10/1/2002	(Standard disk drives)
disk.inf	Not Available				
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_2.38\6&37B414EC&0&17C2					
Emulex Simulate Device	Yes	SYSTEM	5.1.0.0		
2/26/2003	Emulex	oem7.inf	Not Available		
SCSI\ASCIT8&VEN_EMULEX&PROD_UTILITY_IFC_DEV.&REV_E MU2\6&37B414EC&0&27F0					
PCI bus	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard system devices)
machine.inf	Not Available	ACPI\HWP0002\18			
Smart Array 5300 Controller (Non-Miniport)					
	No				
SCSIADAPTER	5.6.59.64	4/8/2003	Hewlett-Packard		
oem14.inf	Not Available				
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&225F72B7&0&08					
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64		
4/8/2003	Hewlett-Packard	oem15.inf	Not		
Available					
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&15C77769&0&0000004000000000					
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64		
4/8/2003	Hewlett-Packard	oem15.inf	Not		
Available					
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&15C77769&0&0100004000000000					
PCI bus	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard system devices)
machine.inf	Not Available	ACPI\HWP0002\1C			
Smart Array 5300 Controller (Non-Miniport)					
	No				
SCSIADAPTER	5.6.59.64	4/8/2003	Hewlett-Packard		
oem14.inf	Not Available				
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&FA5F632&0&08					
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64		
4/8/2003	Hewlett-Packard	oem15.inf	Not		
Available					
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&4F9649B&0&0000004000000000					
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64		
4/8/2003	Hewlett-Packard	oem15.inf	Not		
Available					
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&4F9649B&0&0100004000000000					

Processor	Yes	PROCESSOR	5.2.3790.0	10/1/2002	(Standard processor types)
cpu.inf	Not Available	ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1\9			
Processor Yes PROCESSOR 5.2.3790.0 10/1/2002 (Standard processor types)					
cpu.inf	Not Available	ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1\10			
Processor Yes PROCESSOR 5.2.3790.0 10/1/2002 (Standard processor types)					
cpu.inf	Not Available	ACPI\GENUINEINTEL_IA64_FAMILY_31_MODEL_1\11			
Generic Bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard system devices)					
machine.inf	Not Available	ACPI\HWP0005\2			
Generic Bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard system devices)					
machine.inf	Not Available	ACPI\HWP0004\200			
PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard system devices)					
machine.inf	Not Available	ACPI\HWP0002\40			
HP MP Serial AUX/UPS PortYes PORTS 5.0.3663.16					
12/12/2002	Hewlett-Packard Co.	oem1.inf	Not		
Available					
PCI\VEN_103C&DEV_1290&SUBSYS_1291103C&REV_01\5&194FB0DF&0&00					
HP Management Processor	Yes	SYSTEM	5.0.3790.0		
7/10/2003	Hewlett-Packard Co	oem0.inf	Not		
Available					
PCI\VEN_103C&DEV_1048&SUBSYS_1282103C&REV_03\5&194FB0DF&0&01					
Broadcom NetXtreme Gigabit Ethernet	Yes	NET	7.15.0.0		
8/29/2003	Broadcom	oem2.inf	Not Available		
PCI\VEN_14E4&DEV_1645&SUBSYS_1300103C&REV_15\5&194FB0DF&0&08					
LSI Logic 53C1010-66 Device	Yes	SCSIADAPTER			
5.2.3790.0	10/1/2002	LSI Logic Inc.			
pnpscsi.inf	Not Available				
PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&194FB0DF&0&10					
LSI Logic 53C1010-66 Device	Yes	SCSIADAPTER			
5.2.3790.0	10/1/2002	LSI Logic Inc.			
pnpscsi.inf	Not Available				
PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&194FB0DF&0&11					
LSI Logic 53C1010-66 Device	Yes	SCSIADAPTER			
5.2.3790.0	10/1/2002	LSI Logic Inc.			
pnpscsi.inf	Not Available				
PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&194FB0DF&0&18					

LSI Logic 53C1010-66 Device	Yes	SCSIADAPTER			
5.2.3790.0	10/1/2002	LSI Logic Inc.			
pnpscsi.inf	Not Available				
PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&194FB0DF&0&19					
Smart Array 5300 Controller (Non-Miniport)	No				
SCSIADAPTER	5.6.59.64	4/8/2003	Hewlett-Packard		
oem14.inf	Not Available				
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&548403B&0&08					
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64		
4/8/2003	Hewlett-Packard	oem15.inf	Not		
Available					
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&325B6244&0&0000004000000000					
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64		
4/8/2003	Hewlett-Packard	oem15.inf	Not		
Available					
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&325B6244&0&0100004000000000					
PCI bus	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard system devices)
machine.inf	Not Available	ACPI\HWP0002\4C			
Smart Array 5300 Controller (Non-Miniport)					
	No				
SCSIADAPTER	5.6.59.64	4/8/2003	Hewlett-Packard		
oem14.inf	Not Available				
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&EDA159&0&08					
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64		
4/8/2003	Hewlett-Packard	oem15.inf	Not		
Available					
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&4A2AE8D&0&0000004000000000					
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64		
4/8/2003	Hewlett-Packard	oem15.inf	Not		
Available					
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&4A2AE8D&0&0100004000000000					
Generic Bus	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard system devices)
machine.inf	Not Available	ACPI\HWP0004\201			
PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard system devices)					
machine.inf	Not Available	ACPI\HWP0002\50			

QLogic QLA23xx PCI Fibre Channel Adapter Yes
 SCSIAADAPTER 8.2.2.10 4/16/2003 QLogic
 oem9.inf Not Available
 PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&245
 142EC&0&08
 Qlogic processor device Yes SYSTEM 5.2.3790.0
 10/1/2002 QLOGIC scsdev.inf Not Available
 SCSIAADAPTER\VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
 V_6&2A360411&0&07F0
 PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
 system devices) machine.inf Not Available ACPI\HWP0002\54

 PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
 system devices) machine.inf Not Available ACPI\HWP0002\58

 Smart Array 5300 Controller (Non-Miniport) No
 SCSIAADAPTER 5.6.59.64 4/8/2003 Hewlett-
 Packard oem14.inf Not Available
 PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&A46
 6DE5&0&08
 Smart Array Logical Volume No DISKDRIVE 5.6.56.64
 4/8/2003 Hewlett-Packard oem15.inf Not
 Available
 HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&23
 248CE9&0&0000004000000000
 Smart Array Logical Volume No DISKDRIVE 5.6.56.64
 4/8/2003 Hewlett-Packard oem15.inf Not
 Available
 HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&23
 248CE9&0&0100004000000000
 PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
 system devices) machine.inf Not Available ACPI\HWP0002\5C

 Smart Array 5300 Controller (Non-Miniport) No
 SCSIAADAPTER 5.6.59.64 4/8/2003 Hewlett-
 Packard oem14.inf Not Available
 PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&B10
 CC7&0&08
 Smart Array Logical Volume No DISKDRIVE 5.6.56.64
 4/8/2003 Hewlett-Packard oem15.inf Not
 Available
 HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&D
 419C7E&0&0000004000000000
 Smart Array Logical Volume No DISKDRIVE 5.6.56.64
 4/8/2003 Hewlett-Packard oem15.inf Not
 Available
 HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&D
 419C7E&0&0100004000000000
 Processor Yes PROCESSOR 5.2.3790.0 10/1/2002 (Standard
 processor types) cpu.inf Not Available ACPI\GENUINEINTEL_
 _IA64_FAMILY_31_MODEL_1\12

Processor Yes PROCESSOR 5.2.3790.0 10/1/2002 (Standard
 processor types) cpu.inf Not Available ACPI\GENUINEINTEL_
 _IA64_FAMILY_31_MODEL_1\14
 Processor Yes PROCESSOR 5.2.3790.0 10/1/2002 (Standard
 processor types) cpu.inf Not Available ACPI\GENUINEINTEL_
 _IA64_FAMILY_31_MODEL_1\15
 Generic Bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
 system devices) machine.inf Not Available ACPI\HWP0005\3

 Generic Bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
 system devices) machine.inf Not Available ACPI\HWP0004\300

 PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
 system devices) machine.inf Not Available ACPI\HWP0002\60

 HP MP Serial AUX/UPS Port Yes PORTS 5.0.3663.16
 12/12/2002 Hewlett Packard Co. oem1.inf Not
 Available
 PCI\VEN_103C&DEV_1290&SUBSYS_1291103C&REV_01\5&14C
 294CC&0&00
 HP Management Processor Yes SYSTEM 5.0.3790.0
 7/10/2003 Hewlett-Packard Co oem0.inf Not
 Available
 PCI\VEN_103C&DEV_1048&SUBSYS_1282103C&REV_03\5&14C
 294CC&0&01
 Broadcom NetXtreme Gigabit Ethernet Yes NET 7.15.0.0
 8/29/2003 Broadcom oem2.inf Not Available
 PCI\VEN_14E4&DEV_1645&SUBSYS_1300103C&REV_15\5&14C
 294CC&0&08
 LSI Logic 53C1010-66 Device Yes SCSIAADAPTER
 5.2.3790.0 10/1/2002 LSI Logic Inc.
 pnpscsi.inf Not Available
 PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&14C
 294CC&0&10
 Disk drive Yes DISKDRIVE 5.2.3790.0 10/1/2002 (Standard
 disk drives) disk.inf Not Available
 SCSIAADAPTER\DISK&VEN_HP_36.4G&PROD_ST336706LC&REV_HP05\6&
 1C93B4B9&0&060
 LSI Logic 53C1010-66 Device Yes SCSIAADAPTER
 5.2.3790.0 10/1/2002 LSI Logic Inc.
 pnpscsi.inf Not Available
 PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&14C
 294CC&0&11
 LSI Logic 53C1010-66 Device Yes SCSIAADAPTER
 5.2.3790.0 10/1/2002 LSI Logic Inc.
 pnpscsi.inf Not Available
 PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&14C
 294CC&0&18

LSI Logic 53C1010-66 Device Yes SCSIAADAPTER
 5.2.3790.0 10/1/2002 LSI Logic Inc.
 pnpscsi.inf Not Available
 PCI\VEN_1000&DEV_0021&SUBSYS_1300103C&REV_01\5&14C
 294CC&0&19
 Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
 system devices) machine.inf Not Available ACPI\HWP0002\62

 PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
 system devices) machine.inf Not Available ACPI\HWP0002\64

 PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
 system devices) machine.inf Not Available ACPI\HWP0002\68

 Smart Array 5300 Controller (Non-Miniport) No
 SCSIAADAPTER 5.6.59.64 4/8/2003 Hewlett-
 Packard oem14.inf Not Available
 PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&19D
 51C05&0&08
 Smart Array Logical Volume No DISKDRIVE 5.6.56.64
 4/8/2003 Hewlett-Packard oem15.inf Not
 Available
 HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&16
 BC6346&0&0000004000000000
 Smart Array Logical Volume No DISKDRIVE 5.6.56.64
 4/8/2003 Hewlett-Packard oem15.inf Not
 Available
 HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&16
 BC6346&0&0100004000000000
 PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
 system devices) machine.inf Not Available ACPI\HWP0002\6C

 Smart Array 5300 Controller (Non-Miniport) No
 SCSIAADAPTER 5.6.59.64 4/8/2003 Hewlett-
 Packard oem14.inf Not Available
 PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&155
 1CFD&0&08
 Smart Array Logical Volume No DISKDRIVE 5.6.56.64
 4/8/2003 Hewlett-Packard oem15.inf Not
 Available
 HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&B6
 267C9&0&0000004000000000
 Smart Array Logical Volume No DISKDRIVE 5.6.56.64
 4/8/2003 Hewlett-Packard oem15.inf Not
 Available
 HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&B6
 267C9&0&0100004000000000
 Generic Bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
 system devices) machine.inf Not Available ACPI\HWP0004\301

 PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
 system devices) machine.inf Not Available ACPI\HWP0002\70

QLogic QLA23xx PCI Fibre Channel Adapter	Yes			
SCSIADAPTER	8.2.2.10	4/16/2003	QLogic	
oem9.inf	Not Available			
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&3666E35B&0&08				
Qlogic processor device	Yes	SYSTEM	5.2.3790.0	
10/1/2002	QLOGIC	scsdev.inf	Not Available	
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_02\5&3666E35B&0&08				
V_6&1C366C9A&0&07F0				
PCI bus	Yes	SYSTEM	5.2.3790.0	10/1/2002 (Standard system devices)
system devices)		machine.inf	Not Available	ACPI\HWP0002\74
PCI bus	Yes	SYSTEM	5.2.3790.0	10/1/2002 (Standard system devices)
system devices)		machine.inf	Not Available	ACPI\HWP0002\78
Smart Array 5300 Controller (Non-Miniport)	No			
SCSIADAPTER	5.6.59.64	4/8/2003	Hewlett-Packard	
oem14.inf	Not Available			
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&23CBD7C6&0&08				
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64	
4/8/2003	Hewlett-Packard	oem15.inf	Not	
Available				
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&CF9A53C&0&0000004000000000				
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64	
4/8/2003	Hewlett-Packard	oem15.inf	Not	
Available				
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&CF9A53C&0&0100004000000000				
PCI bus	Yes	SYSTEM	5.2.3790.0	10/1/2002 (Standard system devices)
system devices)		machine.inf	Not Available	ACPI\HWP0002\7C
Smart Array 5300 Controller (Non-Miniport)	No			
SCSIADAPTER	5.6.59.64	4/8/2003	Hewlett-Packard	
oem14.inf	Not Available			
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&2D6138E4&0&08				
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64	
4/8/2003	Hewlett-Packard	oem15.inf	Not	
Available				
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&265EC522&0&0000004000000000				
Smart Array Logical Volume	No	DISKDRIVE	5.6.56.64	
4/8/2003	Hewlett-Packard	oem15.inf	Not	
Available				
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME\6&265EC522&0&0100004000000000				
Logical Disk Manager	Yes	SYSTEM	5.2.3790.0	
10/1/2002	(Standard system devices)	machine.inf	Not	
Available		ROOT\DMIO\0000		

Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&3735C57B&0&LDM#\{CDF596B9-70B1-4F7B-86B0-A2514C95E3}		SYSTEM	5.2.3790.0	
10/1/2002	(Standard system devices)	machine.inf	Not	
Available		ROOT\FTDISK\0000		
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{D6D9362C-9944-4A11-BEE1-E65A2E056B9E}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{9F25A8F6-B91E-4AAD-B720-F6F87B081A00}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{27476FA0-9916-4078-A407-FFB8228C6EE3}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{6E947031-1867-427A-8BFA-8A8038AB49DE}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{4974EFCA-8BFB-440A-BDA8-C0E1BFE0E274}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{A7E55E95-ED70-4384-A10B-0F043586097C}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{C1FD8FA9-50EC-4D81-9438-EACC5AD2B62E}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{9BF4E980-DC1E-4162-9DD8-9C14B1308E4B}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{0E59C695-CEFE-4B93-B06B-0570FAD7ED68}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{04A85DD7-91EB-4674-B89C-EBE61D856152}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{C650FCAD-A95C-42C1-B903-973D9D9D1356}				

Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{F6DCFD6E-829F-C049-B5FC-D9299DCB9947}		VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{35F70E65-D1C4-4784-AA77-2AB2AC8B3EF8}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{FD44D51F-2F50-47C9-969F-8A01B2BAEE96}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{62F165D0-CA5A-4424-9988-2961AB73B171}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{86CE1744-E5E0-47C8-8486-1807DAB98161}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{9D5ACB7D-F2F2-4721-9921-BE77090C9389}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{4CDDE687-8728-4DF8-BD30-36AD373C90B2}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{71041A92-F5BB-4A5B-B859-6CEA5DF26755}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{FA129221-6255-46F3-95CD-04C77954C176}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{F8764954-976F-4589-B9A7-A6CAE8208E3B}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{2279659A-0C7C-4431-8B93-9D190FB7C241}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{2A22D2E9-2005-467A-A9B1-BC71E7E2EC86}				
Generic volume	Yes	VOLUME	5.2.3790.0	
10/1/2002	Microsoft	volume.inf	Not Available	
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{C0E0B715-7CC7-4E04-9DF0-1D2FB86AB753}				

Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CAADDCC6-81BF-4327-AA75-021168DAD9BB}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BD281666-5650-47E2-AB99-8BD4A7DC80F4}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FF838180-4A6E-4534-9A05-9907FA0FCB19}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DE75F49D-36A2-49B5-86DB-22D6E16B264F}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8FD12AEE-3760-4C3B-AF40-E65FAFF47F3D}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5359E755-5A7A-497C-B052-B3DFB89A123E}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{759EEF1E-5DEE-4551-A298-B523BA599D63}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{72091F34-E8BD-4E6F-A9EC-6C798228D3E6}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{87DC5494-EBC0-4508-B8CA-F587A2C4FDD7}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FFD55D0F-948D-4E60-BA66-1CA9355EF919}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{490C7AA4-A195-465B-A5C8-AF3FDEDB73AE}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CBF755E5-00C4-4338-BE72-4FB4918830C7}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{14912822-1FF7-4896-ACB1-180411EB7436}			

Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B8AEC36E-4395-409B-883B-654A1F14FEB3}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{06EE187B-98A5-4F3B-8BAB-85FC7599DDF2}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C267567E-E47B-4AA3-BA53-666A23A1B9E8}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A0D17C18-4EF5-4B1F-A976-E68559053114}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{95D0D43A-75B2-4BCD-B259-090C3F1A4A1F}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{965D010B-F29D-4FB7-84BF-50C440D0F02E}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FC4219D1-4B61-4179-AFDA-95C9B576AEB}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EB419462-CE04-4379-B3F8-65974585A7DD}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F2B71625-3C71-45F1-8B81-DB9C907F6BD7}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CFE4E4D1-077B-4824-AF15-352C14E85AB5}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{873C14C3-DBE1-4D67-8C01-DAC303C6A528}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{44420E3A-00A2-45DF-B52C-22715897980F}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{05EDD379-54C3-4C61-990A-03EF605735D5}			

Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{034E9FE0-4895-409B-883B-654A1F14FEB3}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9A5EB00E-1A05-48D0-B3C9-D8CE38DB7059}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D626CB29-B5C6-4B0D-8D90-73B3E495616D}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0CC9C93F-93C5-48A6-BD54-AA3D5BF80DB1}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{26B3A952-152D-4CC6-AFF0-45E0DA50F95F}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EC4BC685-92BA-4C3A-9F3F-2C1426DAA068}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{89FCD94F-D12B-48B6-B21C-418C36F6C725}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{99C0D402-9B66-4E47-BF54-49A84B1D7786}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{14707E2A-5C60-457F-B476-E123CBC41E5F}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5EF312B3-8A75-404A-941E-856008930E3F}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7414CFBE-0F25-4FD7-8D0A-38CD1555B976}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5796AE27-F1C3-498A-8093-53B4E04EDF14}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{26A55FB1-DE7B-4578-872E-DCB84153BB99}			

Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B5367EBF-7D28-4E0C-921C-1CE87B29D136}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E66B5A11-14E9-4B00-B92F-58E2F5EC51B3}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C21DE8C8-B2DE-4E00-971F-0EF7787C27A4}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{15914B7B-D33C-4B49-A25E-984B88A91BFB}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FA20B75C-CD6C-463C-B617-A793FBAE8E74}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CE662265-4C9E-493E-A6F0-039EC506AA79}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E93966D8-C170-4EE7-9F24-A77C29F9AAC2}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{780E2C0B-D037-4058-A019-8B6BBCB4493A}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D4E33174-907B-49E1-9884-2321A13CC711}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BEB71347-2998-458A-8C3B-C00CC387C24D}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DC6D40E-D-52D6-4185-AE45-C55CB2E676A5}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C6E1DE4-D-A1DE-463A-8EFD-AAB4E96B98CF}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{93F8AA07-C595-4B4A-88CC-422C4B0B4329}			

Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AB588CE9-4E6F-4509-1B-BD6F4088EDC0}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{174B0FD5-7FF8-4623-BFA8-A98190594883}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{01F4C7B5-B3D9-4EE3-8C5D-03E05697468F}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{125F9BD3-F65D-47DD-978E-1B262E1AFFA4}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8245F09C-0D73-4785-A716-470D6DEA298D}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{361C97E8-0CCC-4F71-9C21-00B537DE9BA6}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{85D9127F-050B-4182-BCEB-56F9C8C92065}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5D987B05-4382-4D8B-B7CC-8602A0EE5B63}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{377C1AE3-2A20-4F30-9A90-E2AC630FD035}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4248065D-9F25-46AD-AE42-B62A2F241001}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{01560FA9-E6E8-42EE-8306-2C7C4FDC23D5}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A9AC3287-EFD6-4649-8392-D5014C1A4119}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{706DB7F4-37C5-4F88-97D1-5583842AF2E2}			

Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5B9EF189-0003-089CAF2F8C9}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8A95031D-7681-4289-8154-E43C715B0189}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{292E0348-D43F-4556-883D-EA814D1E38F9}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B751F8C5-64E6-4EC1-B851-B36BFF15B62C}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0D5F273C-5CC1-418F-B13C-3EECAFDF3281}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1177C3C2-4E56-4261-A64A-F5E90BA602D8}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{110A94EB-1F65-49D2-8EAC-4697F7B3A18A}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E4B94800-FFF9-11D7-B7E0-000000000000}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E4B95160-FFF9-11D7-B7E0-000000000000}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E551C774-FFF9-11D7-B7E0-000000000000}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FDC1ED4-8-FFF9-11D7-B7E0-000000000000}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4D612B8-D-5E70-4407-BD1C-577851766408}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4F377555-F7F6-4735-B388-8B742BA929E1}			

Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F3AA8A99-C4ED-44E2-9FCC-6ACC82F59441}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6C2053E0-9577-01C2-92E0-3C772E43AC40}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6C61F0C0-9577-01C2-507B-9E5F8078F531}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{00FD2740-9578-01C2-D931-F8428177D974}			
Generic volume	Yes	VOLUME	5.2.3790.0
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{07827700-9578-01C2-F89F-B474CB61395E}			
AFD Networking Support Environment	Not Available	LEGACYDRIVER	
Available	Not Available	Not Available	Not Available
Beep	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
cpqcssm	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
CRC Disk Filter Driver	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
dmbboot	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
dmload	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
FILEMON	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
Fips	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
Generic Packet Classifier	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
IPSEC driver	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available

ksecdd	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
lpxftr	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
mountmgr	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
NDIS System Driver	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
Remote Access NDIS TAPI Driver	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
NDIS Usermode I/O Protocol	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
NDProxy	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
NetBios over Tcpip	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
Null	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
Partition Manager	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
qlvika	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
Remote Access Auto Connection Driver	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
RDPDCC	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
RDPWD	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
sacdrv	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
TCP/IP Protocol Driver	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
TDTCP	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
volsnap	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available

Remote Access IP ARP Driver	Not Available	LEGACYDRIVER	
Available	Not Available	Not Available	Not Available
Video Codec Key	Not Available	LEGACYDRIVER	Not Available
Available	Not Available	Not Available	Not Available
Legacy Audio Drivers	Yes	MEDIA	5.2.3790.0
10/1/2002	(Standard system devices)	wave.inf	Not
Media Control Devices	Yes	MEDIA	5.2.3790.0
10/1/2002	(Standard system devices)	wave.inf	Not
Legacy Video Capture Devices	Yes	MEDIA	5.2.3790.0
10/1/2002	(Standard system devices)	wave.inf	Not
Video Codecs	Yes	MEDIA	5.2.3790.0
10/1/2002	(Standard system devices)	wave.inf	Not Available
WAN Miniport (L2TP)	Yes	NET	5.2.3790.0
10/1/2002	Microsoft	netrasa.inf	Not Available
WAN Miniport (IP)	Yes	NET	5.2.3790.0
10/1/2002	Microsoft	netrasa.inf	Not Available
WAN Miniport (PPPOE)	Yes	NET	5.2.3790.0
10/1/2002	Microsoft	netrasa.inf	Not Available
WAN Miniport (PPTP)	Yes	NET	5.2.3790.0
10/1/2002	Microsoft	netrasa.inf	Not Available
Direct Parallel	Yes	NET	5.2.3790.0
10/1/2002	Microsoft	netrasa.inf	Not Available
Terminal Server Device Redirector	Yes	SYSTEM	5.2.3790.0
10/1/2002	(Standard system devices)	machine.inf	Not Available
Terminal Server Keyboard Driver	Yes	SYSTEM	5.2.3790.0
10/1/2002	(Standard system devices)	machine.inf	Not Available
Terminal Server Mouse Driver	Yes	SYSTEM	5.2.3790.0
10/1/2002	(Standard system devices)	machine.inf	Not Available
Plug and Play Software Device Enumerator	Yes	SYSTEM	5.2.3790.0
10/1/2002	(Standard system devices)	machine.inf	Not Available

[Environment Variables]

Variable	Value	User Name
ComSpec	%SystemRoot%\system32\cmd.exe	<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\ <SYSTEM>				
windir	%SystemRoot% <SYSTEM>				
OS	Windows_NT <SYSTEM>				
PROCESSOR_ARCHITECTURE	IA64 <SYSTEM>				
PROCESSOR_LEVEL	31 <SYSTEM>				
PROCESSOR_IDENTIFIER	ia64 Family 31 Model 1 Stepping 5, GenuineIntel <SYSTEM>				
PROCESSOR_REVISION	0105 <SYSTEM>				
NUMBER_OF_PROCESSORS	16 <SYSTEM>				
ClusterLog	C:\WINDOWS\Cluster\cluster.log <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 NT AUTHORITY\SYSTEM				
TMP	%USERPROFILE%\Local Settings\Temp NT AUTHORITY\SYSTEM				
TEMP	%USERPROFILE%\Local Settings\Temp NT AUTHORITY\LOCAL SERVICE				
TMP	%USERPROFILE%\Local Settings\Temp NT AUTHORITY\LOCAL SERVICE				
TEMP	%USERPROFILE%\Local Settings\Temp NT AUTHORITY\NETWORK SERVICE				
TMP	%USERPROFILE%\Local Settings\Temp NT AUTHORITY\NETWORK SERVICE				
TEMP	%USERPROFILE%\Local Settings\Temp SQLOLY\Administrator				
TMP	%USERPROFILE%\Local Settings\Temp SQLOLY\Administrator				
[Print Jobs]					
Document Submitted	Size	Owner	Notify	Status	Time
	Start Time	Until Time	Elapsed Time	Pages Printed	
Processor	Job ID	Priority	Parameters	Driver	Print
	Host Print Queue		Data Type	Name	
[Network Connections]					
Local Name	Remote Name	Type	Status	User	
S:	\\ssdllaib\d\$	Disk	Persistent Connection		
	SQLOLY\Administrator				
[Running Tasks]					

Name	Path	Process ID	Priority	Min Working Set	Max Working Set	Start Time	Version	Size
system idle process		0	Not Available	0	Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
system	Not Available	4	8	0	2826240	Not Available	Not Available	Not Available
smss.exe	Not Available	580	11	409600	2826240	12/4/2003 11:48 AM	Not Available	Not Available
Available	Not Available	628	13	Not Available	Not Available	Not Available	Not Available	Not Available
csrss.exe	Not Available	628	13	Not Available	Not Available	Not Available	Not Available	Not Available
Available	12/4/2003 11:48 AM	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
Available	12/4/2003 11:48 AM	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
winlogon.exe	c:\windows\system32\winlogon.exe	652	13	409600	2826240	12/4/2003 11:48 AM	5.2.3790.0 (srv03_rtm.030324-2048)	618.00 KB (632,832)
bytes)	10/16/2003 11:30 AM							
services.exe	c:\windows\system32\services.exe	696	9	409600	2826240	12/4/2003 11:48 AM	5.2.3790.0 (srv03_rtm.030324-2048)	286.00 KB (292,864)
bytes)	10/16/2003 11:30 AM							
lsass.exe	c:\windows\system32\lsass.exe	708	9	409600	2826240	12/4/2003 11:48 AM	5.2.3790.0 (srv03_rtm.030324-2048)	15.00 KB (15,360)
bytes)	9/18/2003 11:53 AM							
svchost.exe	c:\windows\system32\svchost.exe	880	8	409600	2826240	12/4/2003 11:48 AM	5.2.3790.0 (srv03_rtm.030324-2048)	32.50 KB (33,280)
bytes)	9/18/2003 11:53 AM							
svchost.exe	c:\windows\system32\svchost.exe	924	8	409600	2826240	12/4/2003 11:48 AM	5.2.3790.0 (srv03_rtm.030324-2048)	32.50 KB (33,280)
bytes)	9/18/2003 11:53 AM							
svchost.exe	Not Available	1060	8	Not Available	Not Available	Not Available	Not Available	Not Available
Available	12/4/2003 11:48 AM	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
Available	12/4/2003 11:48 AM	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
svchost.exe	Not Available	1112	8	Not Available	Not Available	Not Available	Not Available	Not Available
Available	12/4/2003 11:48 AM	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
Available	12/4/2003 11:48 AM	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
svchost.exe	c:\windows\system32\svchost.exe	1160	8	409600	2826240	12/4/2003 11:48 AM	5.2.3790.0 (srv03_rtm.030324-2048)	32.50 KB (33,280)
bytes)	9/18/2003 11:53 AM							
msdtc.exe	Not Available	1256	8	Not Available	Not Available	Not Available	Not Available	Not Available
Available	12/4/2003 11:48 AM	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
Available	12/4/2003 11:48 AM	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
dfssvc.exe	c:\windows\system32\dfssvc.exe	1844	8	409600	2826240	12/4/2003 11:48 AM	5.2.3790.0 (srv03_rtm.030324-2048)	442.00 KB (452,608)
bytes)	10/16/2003 11:31 AM							
wmiprvse.exe	Not Available	428	8	Not Available	Not Available	Not Available	Not Available	Not Available
Available	Not Available	12/4/2003 11:49 AM	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available

csrss.exe	Not Available	252	13	Not Available	Not Available	Not Available	Not Available	Not Available
Available	12/5/2003 2:15 PM	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
Available	12/5/2003 2:15 PM	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
winlogon.exe	c:\windows\system32\winlogon.exe	1272	13	409600	2826240	12/5/2003 2:15 PM	5.2.3790.0 (srv03_rtm.030324-2048)	618.00 KB (632,832)
bytes)	10/16/2003 11:30 AM							
logon.scr	Not Available	2276	4	Not Available	Not Available	Not Available	Not Available	Not Available
Available	12/5/2003 2:25 PM	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
Available	12/5/2003 2:25 PM	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
csrss.exe	Not Available	2352	13	Not Available	Not Available	Not Available	Not Available	Not Available
Available	12/5/2003 2:30 PM	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
Available	12/5/2003 2:30 PM	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
winlogon.exe	c:\windows\system32\winlogon.exe	2380	13	409600	2826240	12/5/2003 2:30 PM	5.2.3790.0 (srv03_rtm.030324-2048)	618.00 KB (632,832)
bytes)	10/16/2003 11:30 AM							
rdpclip.exe	c:\windows\system32\rdpclip.exe	2540	8	409600	2826240	12/5/2003 2:30 PM	5.2.3790.0 (srv03_rtm.030324-2048)	156.00 KB (159,744)
bytes)	10/16/2003 11:30 AM							
explorer.exe	c:\windows\explorer.exe	2624	8	409600	2826240	12/5/2003 2:30 PM	5.2.3790.0 (srv03_rtm.030324-2048)	1.63 MB (1,704,960 bytes)
bytes)	9/18/2003 11:53 AM							
jusched.exe	c:\program files (x86)\java\j2re1.4.2_01\bin\jusched.exe	2708	8	409600	2826240	12/5/2003 2:30 PM	8/19/2007 5:23 PM	32.10 KB (32,873)
bytes)	8/19/2007 5:23 PM							
rdpclip.exe	c:\windows\system32\rdpclip.exe	588	8	409600	2826240	12/8/2003 10:29 AM	5.2.3790.0 (srv03_rtm.030324-2048)	156.00 KB (159,744)
bytes)	10/16/2003 11:30 AM							
explorer.exe	c:\windows\explorer.exe	548	8	409600	2826240	12/8/2003 10:29 AM	5.2.3790.0 (srv03_rtm.030324-2048)	1.63 MB (1,704,960 bytes)
bytes)	9/18/2003 11:53 AM							
jusched.exe	c:\program files (x86)\java\j2re1.4.2_01\bin\jusched.exe	396	8	409600	2826240	12/8/2003 10:29 AM	8/19/2007 5:23 PM	32.10 KB (32,873)
bytes)	8/19/2007 5:23 PM							
cmd.exe	c:\windows\system32\cmd.exe	2000	8	409600	2826240	12/8/2003 10:29 AM	5.2.3790.0 (srv03_rtm.030324-2048)	493.50 KB (505,344)
bytes)	10/16/2003 11:30 AM							
cmd.exe	c:\windows\system32\cmd.exe	3768	8	409600	2826240	12/8/2003 4:06 PM	5.2.3790.0 (srv03_rtm.030324-2048)	493.50 KB (505,344)
bytes)	10/16/2003 11:30 AM							
cmd.exe	c:\windows\system32\cmd.exe	424	8	409600	2826240	12/8/2003 4:32 PM	5.2.3790.0 (srv03_rtm.030324-2048)	493.50 KB (505,344)
bytes)	10/16/2003 11:30 AM							

```

helpctr.exe c:\windows\pchealth\helpctr\binaries\helpctr.exe 4048
8 409600 2826240 12/8/2003 4:32 PM
5.2.3790.0 (srv03_rtm.030324-2048) 1.97 MB (2,066,432
bytes) 9/18/2003 12:14 PM
wmiprvse.exe Not Available 2552 8 Not
Available Not Available 12/8/2003 4:32 PM
Available Not Available
helpsvc.exe c:\windows\pchealth\helpctr\binaries\helpsvc.exe 3268
8 409600 2826240 12/8/2003 4:32 PM
5.2.3790.0 (srv03_rtm.030324-2048) 2.18 MB (2,289,152
bytes) 10/16/2003 11:31 AM

```

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer Path
winlogon bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	618.00 KB (632,832 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
ntdll bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	1.45 MB (1,524,224 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
kernel32 bytes)	5.2.3790.73 (srv03_gfe.030614-1407)	1.76 MB (1,850,368 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
msvcrt bytes)	7.0.3790.0 (srv03_rtm.030324-2048)	873.50 KB (894,464 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
advapi32 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	1.32 MB (1,383,424 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
rpcrt4 bytes)	5.2.3790.76 (srv03_gdr.030808-1703)	2.03 MB (2,127,872 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
user32 bytes)	5.2.3790.73 (srv03_gdr.030614-1005)	1.31 MB (1,372,672 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
gdi32 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	783.00 KB (801,792 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
userenv bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	1.46 MB (1,536,000 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
nddeapi bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	39.50 KB (40,448 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
crypt32 bytes)	5.131.3790.0 (srv03_rtm.030324-2048)	1.50 MB (1,576,448 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
msasn1 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	153.50 KB (157,184 bytes)	10/16/2003 11:30 AM	Microsoft Corporation

```

secur32 5.2.3790.0 (srv03_rtm.030324-2048) 166.50 KB (170,496
bytes) 10/16/2003 11:30 AM Microsoft Corporation
winsta 5.2.3790.0 (srv03_rtm.030324-2048) 138.50 KB (141,824
bytes) 10/16/2003 11:30 AM Microsoft Corporation
netapi32 c:\windows\system32\netapi32.dll 832.00 KB (851,968
bytes) 10/16/2003 11:30 AM Microsoft Corporation
profmap c:\windows\system32\profmap.dll 55.50 KB (56,832
bytes) 9/18/2003 11:53 AM Microsoft Corporation
regapi c:\windows\system32\regapi.dll 124.00 KB (126,976
bytes) 10/16/2003 11:30 AM Microsoft Corporation
ws2_32 5.2.3790.0 (srv03_rtm.030324-2048) 228.50 KB (233,984
bytes) 10/16/2003 11:30 AM Microsoft Corporation
ws2help c:\windows\system32\ws2_32.dll 49.50 KB (50,688
bytes) 9/18/2003 11:53 AM Microsoft Corporation
msgina c:\windows\system32\msgina.dll 1.35 MB (1,417,728
bytes) 10/16/2003 11:30 AM Microsoft Corporation
shsvcs 6.00.3790.0 (srv03_rtm.030324-2048) 321.50 KB (329,216
bytes) 9/18/2003 11:53 AM Microsoft Corporation
shlwapi 6.00.3790.0 (srv03_rtm.030324-2048) 722.00 KB (739,328
bytes) 9/18/2003 11:53 AM Microsoft Corporation
sfc c:\windows\system32\sfc.dll 7.50 KB (7,680 bytes)
9/18/2003 11:53 AM Microsoft Corporation
sfc_os c:\windows\system32\sfc_os.dll 257.00 KB (263,168
bytes) 10/16/2003 11:30 AM Microsoft Corporation
wintrust 5.131.3790.0 (srv03_rtm.030324-2048) 451.50 KB (462,336
bytes) 9/18/2003 11:53 AM Microsoft Corporation
ole32 c:\windows\system32\ole32.dll 3.39 MB (3,551,744
bytes) 10/16/2003 11:30 AM Microsoft Corporation
imagehlp 5.2.3790.0 (srv03_rtm.030324-2048) 128.50 KB (131,584
bytes) 9/18/2003 11:53 AM Microsoft Corporation
comctl32 c:\windows\system32\imagehlp.dll 2.18 MB (2,285,056
bytes) 6.0 (srv03_rtm.030324-2048)
9/18/2003 5:00 AM Microsoft Corporation
controls_6595b64144ccf1df_6.0.100.0_x-ww_b3722bab\comctl32.dll
c:\windows\winsxs\ia64_microsoft.windows.common-
version 5.2.3790.0 (srv03_rtm.030324-2048) 44.00 KB (45,056
bytes) 9/18/2003 11:53 AM Microsoft Corporation
c:\windows\system32\version.dll

```

```

wincard 5.2.3790.0 (srv03_rtm.030324-2048) 291.50 KB (298,496
bytes) 9/18/2003 11:53 AM Microsoft Corporation
wtsapi32 5.2.3790.0 (srv03_rtm.030324-2048) 47.50 KB (48,640
bytes) 9/18/2003 11:53 AM Microsoft Corporation
sxs c:\windows\system32\wtsapi32.dll 1.77 MB (1,860,608
bytes) 9/18/2003 11:53 AM Microsoft Corporation
winmm c:\windows\system32\sxs.dll 404.00 KB (413,696
bytes) 9/18/2003 11:53 AM Microsoft Corporation
shell32 6.00.3790.0 (srv03_rtm.030324-2048) 12.35 MB (12,953,088
bytes) 10/16/2003 11:30 AM Microsoft Corporation
setupapi c:\windows\system32\shell32.dll 1.90 MB (1,991,168
bytes) 9/18/2003 11:53 AM Microsoft Corporation
wldap32 c:\windows\system32\setupapi.dll 412.50 KB (422,400
bytes) 10/16/2003 11:30 AM Microsoft Corporation
cscdll c:\windows\system32\wldap32.dll 211.00 KB (216,064
bytes) 10/16/2003 11:30 AM Microsoft Corporation
winotify c:\windows\system32\cscdll.dll 218.00 KB (223,232
bytes) 9/18/2003 11:53 AM Microsoft Corporation
winspool 5.2.3790.0 (srv03_rtm.030324-2048) 399.50 KB (409,088
bytes) 10/16/2003 11:30 AM Microsoft Corporation
mpr c:\windows\system32\winspool.drv 163.00 KB (166,912
bytes) 9/18/2003 11:53 AM Microsoft Corporation
comctl32 5.82 (srv03_rtm.030324-2048) 1.55 MB (1,621,504
bytes) 9/18/2003 5:00 AM Microsoft Corporation
controls_6595b64144ccf1df_5.82.0.0_x-ww_b9c4a0a5\comctl32.dll
c:\windows\winsxs\ia64_microsoft.windows.common-
uxtheme 6.00.3790.0 (srv03_rtm.030324-2048) 527.50 KB (540,160
bytes) 10/16/2003 11:30 AM Microsoft Corporation
clbcatq 2001.12.4720.0 (srv03_rtm.030324-2048) 1.23 MB
(1,292,800 bytes) 9/18/2003 12:11 PM Microsoft Corporation
oleaut32 c:\windows\system32\clbcatq.dll 3.57 MB (3,739,136 bytes)
9/18/2003 11:53 AM Microsoft Corporation
comres 2001.12.4720.0 (srv03_rtm.030324-2048) 779.50
KB (798,208 bytes) 9/18/2003 11:53 AM Microsoft Corporation
wbemprox c:\windows\system32\comres.dll 46.00 KB (47,104
bytes) 10/16/2003 11:31 AM Microsoft Corporation
c:\windows\system32\wbem\wbemprox.dll

```

wbemcomn bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	598.50 KB (612,864 bytes)	10/16/2003 11:31 AM	Microsoft Corporation
	c:\windows\system32\wbem\wbemcomn.dll			
wbemsvc bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	62.50 KB (64,000 bytes)	9/18/2003 12:11 PM	Microsoft Corporation
	c:\windows\system32\wbem\wbemsvc.dll			
fastprox bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	1.51 MB (1,580,544 bytes)	10/16/2003 11:31 AM	Microsoft Corporation
	c:\windows\system32\wbem\fastprox.dll			
msvcfp60	6.10.2240.8	941.50 KB (964,096 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\msvcfp60.dll			
ntdsapi bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	181.50 KB (185,856 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\ntdsapi.dll			
dnsapi bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	404.00 KB (413,696 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\dnsapi.dll			
scredir bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	67.50 KB (69,120 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\scredir.dll			
samlib bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	106.00 KB (108,544 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\samlib.dll			
drprov bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	26.50 KB (27,136 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\drprov.dll			
rdpsnd bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	63.00 KB (64,512 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\rdpsnd.dll			
psapi bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	48.00 KB (49,152 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\psapi.dll			
ntlanman bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	108.00 KB (110,592 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\ntlanman.dll			
netui0 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	181.50 KB (185,856 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\netui0.dll			
netui1 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	482.00 KB (493,568 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\netui1.dll			
davclnt bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	59.00 KB (60,416 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\davclnt.dll			
mprui bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	97.00 KB (99,328 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\mprui.dll			
netui2 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	761.50 KB (779,776 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\netui2.dll			

comdlg32 bytes)	6.00.3790.0 (srv03_rtm.030324-2048)	706.00 KB (722,944 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
netmsg bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	177.50 KB (181,760 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\netmsg.dll			
msacm32 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	77.00 KB (78,848 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\msacm32.drv			
msacm32 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	252.00 KB (258,048 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\msacm32.dll			
es	2001.12.4720.0 (srv03_rtm.030324-2048)	637.50 KB (652,800 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\es.dll			
ntmarta bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	343.50 KB (351,744 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\ntmarta.dll			
imaadp32 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	55.00 KB (56,320 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\imaadp32.acm			
msadp32 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	49.00 KB (50,176 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\msadp32.acm			
msg711 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	33.00 KB (33,792 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\msg711.acm			
msgsm32 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	66.50 KB (68,096 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\msgsm32.acm			
tssoft32	1.01	29.00 KB (29,696 bytes)	10/16/2003 11:30 AM	DSP GROUP, INC.
	c:\windows\system32\tssoft32.acm			
tsd32	Not Available	38.00 KB (38,912 bytes)	10/16/2003 11:30 AM	
	c:\windows\system32\tsd32.dll			
authz bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	202.50 KB (207,360 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\authz.dll			
cryptnet bytes)	5.131.3790.0 (srv03_rtm.030324-2048)	153.50 KB (157,184 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\cryptnet.dll			
sensapi bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	11.50 KB (11,776 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\sensapi.dll			
scgntfy bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	42.50 KB (43,520 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\scgntfy.dll			
rasapi32 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	589.50 KB (603,648 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\rasapi32.dll			
rasman bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	154.50 KB (158,208 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\rasman.dll			

tapi32 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	493.00 KB (504,832 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
rtutils bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	81.50 KB (83,456 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\rtutils.dll			
cscui bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	574.00 KB (587,776 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\cscui.dll			
services bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	286.00 KB (292,864 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\services.exe			
scesrv bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	765.00 KB (783,360 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\scesrv.dll			
umpnpgmr bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	314.50 KB (322,048 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\umpnpgmr.dll			
ncobjapi bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	118.50 KB (121,344 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\ncobjapi.dll			
eventlog bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	157.00 KB (160,768 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\eventlog.dll			
lsass bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	15.00 KB (15,360 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\lsass.exe			
lsasrv bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	1.94 MB (2,033,664 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\lsasrv.dll			
samsrv (1,029,120 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	1,005.00 KB	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\samsrv.dll			
cryptdll bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	61.00 KB (62,464 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\cryptdll.dll			
msprivs bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	46.00 KB (47,104 bytes)	9/18/2003 11:53 AM	Microsoft Corporation
	c:\windows\system32\msprivs.dll			
kerberos bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	876.00 KB (897,024 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\kerberos.dll			
msv1_0 bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	333.50 KB (341,504 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\msv1_0.dll			
netlogon bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	936.50 KB (958,976 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\netlogon.dll			
w32time bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	540.50 KB (553,472 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\w32time.dll			
iphlpapi bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	223.00 KB (228,352 bytes)	10/16/2003 11:30 AM	Microsoft Corporation
	c:\windows\system32\iphlpapi.dll			

schannel bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	468.50 KB (479,744 bytes)
wdigest bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	161.50 KB (165,376 bytes)
rsaenh bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	371.83 KB (380,752 bytes)
rassfm bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	56.00 KB (57,344 bytes)
kdcsvc bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	571.50 KB (585,216 bytes)
ntdsa bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	3.82 MB (4,008,448 bytes)
ntdsatq bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	80.50 KB (82,432 bytes)
mswsock bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	671.00 KB (687,104 bytes)
esent bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	2.48 MB (2,605,056 bytes)
scecli bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	467.50 KB (478,720 bytes)
wshtcpip bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	38.00 KB (38,912 bytes)
ipsecsvc bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	410.50 KB (420,352 bytes)
oakley bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	493.50 KB (505,344 bytes)
winipsec bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	78.50 KB (80,384 bytes)
pstorsvc bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	56.00 KB (57,344 bytes)
psbase bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	162.50 KB (166,400 bytes)
dsenh bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	319.33 KB (326,992 bytes)

wlbsctrl bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	194.50 KB (199,168 bytes)
svchost bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	32.50 KB (33,280 bytes)
rpcss bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	650.00 KB (665,600 bytes)
termsrv bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	607.00 KB (621,568 bytes)
icaapi bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	27.00 KB (27,648 bytes)
mstlsapi bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	311.00 KB (318,464 bytes)
activeds bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	543.50 KB (556,544 bytes)
adslidpc bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	312.00 KB (319,488 bytes)
credui bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	288.00 KB (294,912 bytes)
atl	3.00.2282	348.00 KB (356,352 bytes)
rdpwsx bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	256.63 KB (262,792 bytes)
schedsvc bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	527.50 KB (540,160 bytes)
msidle	6.00.3790.0 (srv03_rtm.030324-2048)	8.50 KB (8,704 bytes)
wkssvc bytes)	5.2.3790.90 (srv03_gdr.030906-2150)	304.00 KB (311,296 bytes)
cryptsvc bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	126.00 KB (129,024 bytes)
certcli bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	586.00 KB (600,064 bytes)
vssapi bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	1.28 MB (1,339,904 bytes)
dmserver bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	45.50 KB (46,592 bytes)

srsvc bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	188.00 KB (192,512 bytes)
sacsvr bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	27.50 KB (28,160 bytes)
sens bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	90.50 KB (92,672 bytes)
comsvcs (3,106,816 bytes)	2001.12.4720.0 (srv03_rtm.030324-2048)	2.96 MB
trkwks bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	246.00 KB (251,904 bytes)
wmisvc bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	408.50 KB (418,304 bytes)
browser bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	187.00 KB (191,488 bytes)
netrap bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	30.00 KB (30,720 bytes)
wiarpc bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	67.50 KB (69,120 bytes)
wbemcore bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	1.64 MB (1,723,904 bytes)
esscli bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	919.50 KB (941,568 bytes)
wmiutils bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	285.00 KB (291,840 bytes)
repdrvfs bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	595.50 KB (609,792 bytes)
wmiprvsd bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	1.38 MB (1,443,328 bytes)
wbemess bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	976.50 KB (999,936 bytes)
ncprov bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	133.50 KB (136,704 bytes)
ntlsapi bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	14.50 KB (14,848 bytes)
xactsrv bytes)	5.2.3790.0 (srv03_rtm.030324-2048)	283.50 KB (290,304 bytes)

netman bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\netman.dll	591.00 KB (605,184 bytes)
mprapi bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\mprapi.dll	238.00 KB (243,712 bytes)
wzcsvc bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\wzcsvc.dll	608.50 KB (623,104 bytes)
wmi	5.2.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\wmi.dll	5.00 KB (5,120 bytes)
dhcpcsvc bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\dhcpcsvc.dll	279.50 KB (286,208 bytes)
wzcsapi bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\wzcsapi.dll	49.50 KB (50,688 bytes)
netshell bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\netshell.dll	2.65 MB (2,779,648 bytes)
clusapi bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\clusapi.dll	165.50 KB (169,472 bytes)
hnetcfg bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\hnetcfg.dll	764.00 KB (782,336 bytes)
wininet bytes)	6.00.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\wininet.dll	1.43 MB (1,500,672 bytes)
rasdlg bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\rasdlg.dll	1.35 MB (1,420,800 bytes)
rasadhlp bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\rasadhlp.dll	13.00 KB (13,312 bytes)
winmr bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\winmr.dll	39.00 KB (39,936 bytes)
pchsvc bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 9/18/2003 12:14 PM Microsoft Corporation c:\windows\system32\pchealth\helpctr\binaries\pchsvc.dll	94.50 KB (96,768 bytes)
wbemcons bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:31 AM Microsoft Corporation c:\windows\system32\wbem\wbemcons.dll	237.50 KB (243,200 bytes)
dfssvc bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:31 AM Microsoft Corporation c:\windows\system32\dfssvc.exe	442.00 KB (452,608 bytes)
resutils bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\resutils.dll	147.50 KB (151,040 bytes)

mfc42u	6.00.3014.0 3.34 MB (3,506,176 bytes) Microsoft Corporation c:\windows\system32\mfc42u.dll	9/18/2003 11:53 AM
wsock32 bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\wsock32.dll	23.00 KB (23,552 bytes)
rdpclip bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\rdpclip.exe	156.00 KB (159,744 bytes)
explorer bytes)	6.00.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\explorer.exe	1.63 MB (1,704,960 bytes)
browseui bytes)	6.00.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\browseui.dll	2.42 MB (2,536,960 bytes)
shdocvw bytes)	6.00.3790.85 (srv03_gdr.030906-2150) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\shdocvw.dll	3.20 MB (3,359,744 bytes)
apphelp bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\apphelp.dll	262.50 KB (268,800 bytes)
themeui bytes)	6.00.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\themeui.dll	823.00 KB (842,752 bytes)
msimg32	5.2.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\msimg32.dll	7.00 KB (7,168 bytes)
actxprxy bytes)	6.00.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:52 AM Microsoft Corporation c:\windows\system32\actxprxy.dll	230.00 KB (235,520 bytes)
linkinfo bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\linkinfo.dll	42.00 KB (43,008 bytes)
ntshrui bytes)	6.00.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\ntshrui.dll	233.50 KB (239,104 bytes)
webcheck bytes)	6.00.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\webcheck.dll	664.50 KB (680,448 bytes)
stobject bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\stobject.dll	178.00 KB (182,272 bytes)
batmeter bytes)	6.00.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\batmeter.dll	55.50 KB (56,832 bytes)
powrprof bytes)	6.00.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\powrprof.dll	36.00 KB (36,864 bytes)
printui bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\printui.dll	1.09 MB (1,142,784 bytes)
cfgmgr32 bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\cfgmgr32.dll	16.00 KB (16,384 bytes)

jusched	Not Available 32.10 KB (32,873 bytes) Not Available c:\program files c:\windows\system32\jusched.exe	8/19/2007 5:23 PM
java2s bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\java2s.dll	438.50 KB (449,024 bytes)
wow64win bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\wow64win.dll	514.50 KB (526,848 bytes)
wow64cpu bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\wow64cpu.dll	38.00 KB (38,912 bytes)
urlmon bytes)	6.00.3790.87 (srv03_gdr.030906-2150) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\urlmon.dll	1.21 MB (1,271,808 bytes)
browsecl bytes)	6.00.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\browsecl.dll	61.50 KB (62,976 bytes)
shdoclc bytes)	6.00.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\shdoclc.dll	588.00 KB (602,112 bytes)
mydocs bytes)	6.00.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\mydocs.dll	129.00 KB (132,096 bytes)
sendmail bytes)	6.00.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\sendmail.dll	97.00 KB (99,328 bytes)
cmd bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\cmd.exe	493.50 KB (505,344 bytes)
helpctr bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 9/18/2003 12:14 PM Microsoft Corporation c:\windows\system32\pchealth\helpctr\binaries\helpctr.exe	1.97 MB (2,066,432 bytes)
hcappres bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 9/18/2003 12:14 PM Microsoft Corporation c:\windows\system32\pchealth\helpctr\binaries\hcappres.dll	6.00 KB (6,144 bytes)
itss bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\itss.dll	349.00 KB (357,376 bytes)
msxml3	8.40.9419.0 3.42 MB (3,590,656 bytes) Microsoft Corporation c:\windows\system32\msxml3.dll	10/16/2003 11:30 AM
pchshell bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 9/18/2003 12:14 PM Microsoft Corporation c:\windows\system32\pchealth\helpctr\binaries\pchshell.dll	277.50 KB (284,160 bytes)
mlang bytes)	6.00.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\mlang.dll	799.00 KB (818,176 bytes)
mshtml bytes)	6.00.3790.88 (srv03_gdr.030906-2150) 10/16/2003 11:30 AM Microsoft Corporation c:\windows\system32\mshtml.dll	7.83 MB (8,210,944 bytes)
msimtf bytes)	5.2.3790.0 (srv03_rtm.030324-2048) 9/18/2003 11:53 AM Microsoft Corporation c:\windows\system32\msimtf.dll	528.00 KB (540,672 bytes)

mctf	5.2.3790.0 (srv03_rtm.030324-2048)	924.50 KB (946,688 bytes)	10/16/2003 11:30 AM	Microsoft Corporation	c:\windows\system32\mctf.dll
jscript	5.6.0.8515	1.21 MB (1,268,736 bytes)	9/18/2003 11:53 AM	Microsoft Corporation	c:\windows\system32\jscrip.dll
msls31	3.10.349.0	448.00 KB (458,752 bytes)	9/18/2003 11:53 AM	Microsoft Corporation	c:\windows\system32\msls31.dll
imm32	5.2.3790.0 (srv03_rtm.030324-2048)	307.50 KB (314,880 bytes)	9/18/2003 11:53 AM	Microsoft Corporation	c:\windows\system32\imm32.dll
mshtml	6.00.3790.0 (srv03_rtm.030324-2048)	1.34 MB (1,408,512 bytes)	9/18/2003 11:53 AM	Microsoft Corporation	c:\windows\system32\mshtml.dll
vbscript	5.6.0.8515	1.06 MB (1,110,016 bytes)	9/18/2003 11:53 AM	Microsoft Corporation	c:\windows\system32\vbscript.dll
mfc42	6.00.3014.0	3.36 MB (3,526,656 bytes)	9/18/2003 11:53 AM	Microsoft Corporation	c:\windows\system32\mfc42.dll
msinfo	5.2.3790.0 (srv03_rtm.030324-2048)	1.20 MB (1,257,984 bytes)	9/18/2003 12:14 PM	Microsoft Corporation	c:\windows\pchealth\helpctr\binaries\msinfo.dll
riched32	5.2.3790.0 (srv03_rtm.030324-2048)	5.00 KB (5,120 bytes)	9/18/2003 11:53 AM	Microsoft Corporation	c:\windows\system32\riched32.dll
riched20	5.31.23.1218	1.25 MB (1,313,280 bytes)	9/18/2003 11:53 AM	Microsoft Corporation	c:\windows\system32\riched20.dll
helpsvc	5.2.3790.0 (srv03_rtm.030324-2048)	2.18 MB (2,289,152 bytes)	10/16/2003 11:31 AM	Microsoft Corporation	c:\windows\pchealth\helpctr\binaries\helpsvc.exe

[Services]

Display Name	Name	State	Start Mode	Service
Type	Path	Error Control	Start Name	Tag ID
Alert	Alert	Stopped	Disabled	Share Process
	c:\windows\system32\svchost.exe -k localservice			Normal
	NT AUTHORITY\LocalService		0	
Application Layer Gateway Service	ALG	Stopped		Manual
	Own Process c:\windows\system32\alg.exe			Normal
	NT AUTHORITY\LocalService		0	
Application Management	AppMgmt	Stopped		Manual
Process	c:\windows\system32\svchost.exe -k netsvcs			Normal
	LocalSystem 0			
Windows Audio	AudioSrv	Stopped	Disabled	Share
Process	c:\windows\system32\svchost.exe -k netsvcs			Normal
	LocalSystem 0			

Background Intelligent Transfer Service	BITS	Stopped		Manual
	Share Process			
	c:\windows\system32\svchost.exe -k			
Computer Browser	Browser	Running	Auto	Share
Process	c:\windows\system32\svchost.exe -k netsvcs			Normal
	LocalSystem 0			
Indexing Service	CISvc	Stopped	Disabled	Share
Process	c:\windows\system32\cisvc.exe			Normal
	LocalSystem 0			
ClipBook	ClipSrv	Stopped	Disabled	Own Process
	c:\windows\system32\clipsrv.exe			Normal
	LocalSystem 0			
COM+ System Application	COMSysApp	Stopped		Manual
Process	c:\windows\system32\dlhst.exe /processid:{02d4b3f1-fd88-11d1-960d-00805fc79235}			LocalSystem 0
Cryptographic Services	CryptSvc	Running	Auto	Share
Process	c:\windows\system32\svchost.exe -k netsvcs			Normal
	LocalSystem 0			
Distributed File System	Dfs	Running	Auto	Own
Process	c:\windows\system32\dfssvc.exe			Normal
	LocalSystem 0			
DHCP Client	Dhcp	Running	Auto	Share Process
	c:\windows\system32\svchost.exe -k networkservice			Normal
	NT AUTHORITY\NetworkService		0	
Logical Disk Manager	Administrative Service			dmadmin
	Manual			Share Process
	c:\windows\system32\dmadmin.exe /com			Normal
	LocalSystem 0			
Logical Disk Manager	dmserver	Running	Auto	Share
Process	c:\windows\system32\svchost.exe -k netsvcs			Normal
	LocalSystem 0			
DNS Client	Dnscache	Running	Auto	Share Process
	c:\windows\system32\svchost.exe -k networkservice			Normal
	NT AUTHORITY\NetworkService		0	
Error Reporting Service	ERSvc	Stopped	Disabled	Share
Process	c:\windows\system32\svchost.exe -k winerr			Ignore
	LocalSystem 0			
Event Log	Eventlog	Running	Auto	Share Process
	c:\windows\system32\services.exe			Normal
	LocalSystem 0			
COM+ Event System	EventSystem	Running		Manual
Process	c:\windows\system32\svchost.exe -k netsvcs			Share
	LocalSystem 0			Normal
Help and Support	helpsvc	Running		Manual
Process	c:\windows\system32\svchost.exe -k netsvcs			Share
	LocalSystem 0			Normal
Human Interface Device Access	HidServ	Stopped		Disabled
	Share Process			
	c:\windows\system32\svchost.exe -k			
netsvcs	Normal			LocalSystem 0
HTTP SSL	HTTPFilter	Stopped		Manual
	c:\windows\system32\lsass.exe			Share Process
	LocalSystem 0			Normal

IAS Jet Database Access	IASJet	Stopped		Manual
Process	c:\windows\system32\svchost.exe -k iasjet			Share
IMAPI CD-Burning COM Service	ImapiService	Stopped		Normal
	Own Process c:\windows\system32\imapi.exe			Disabled
	LocalSystem 0			Normal
Intersite Messaging	IsmServ	Stopped		Disabled
Process	c:\windows\system32\ismserv.exe			Own
	LocalSystem 0			Normal
Kerberos Key Distribution Center	kdc	Stopped		Disabled
	Share Process			
	c:\windows\system32\lsass.exe			
	Normal			LocalSystem 0
Server	lanmanserverRunning	Auto		Share Process
	c:\windows\system32\svchost.exe -k netsvcs			Normal
	LocalSystem 0			
Workstation	lanmanworkstation	Running	Auto	Share
Process	c:\windows\system32\svchost.exe -k netsvcs			Normal
	LocalSystem 0			
License Logging	LicenseService	Stopped		Disabled
	Own Process c:\windows\system32\llssrv.exe			Normal
	NT AUTHORITY\NetworkService		0	
TCP/IP NetBIOS Helper	LmHosts	Running	Auto	Share
Process	c:\windows\system32\svchost.exe -k localservice			Normal
	NT AUTHORITY\LocalService		0	
Messenger	Messenger	Stopped	Disabled	Share Process
	c:\windows\system32\svchost.exe -k netsvcs			Normal
	LocalSystem 0			
Distributed Transaction Coordinator	MSDTC	Running		Auto
	Own Process c:\windows\system32\msdtc.exe			Normal
	NT AUTHORITY\NetworkService		0	
Windows Installer	MSIServer	Stopped		Manual
Process	c:\windows\system32\msiexec.exe /v			Share
	LocalSystem 0			Normal
MSSQLSERVER	MSSQLSERVER	Stopped		Disabled
	Own Process c:\program files\microsoft sql			
server\mssql\binn\sqlservr.exe -smssqlserver				Normal
	LocalSystem 0			
MSSQLServerADHelper	MSSQLServerADHelper	Stopped		Manual
	Own Process c:\program files\microsoft sql			
server\80\tools\binn\sqladhip.exe				Normal
	LocalSystem 0			
Network DDE	NetDDE	Stopped	Disabled	Share Process
	c:\windows\system32\ntddde.exe			Normal
	LocalSystem 0			
Network DDE DSDM	NetDDEdsdm	Stopped		Disabled
Process	c:\windows\system32\ntddde.exe			Share
	LocalSystem 0			Normal
Net Logon	Netlogon	Stopped		Manual
	c:\windows\system32\lsass.exe			Share Process
	LocalSystem 0			Normal
Network Connections	Netman	Running		Manual
Process	c:\windows\system32\svchost.exe -k netsvcs			Share
	LocalSystem 0			Normal

Network Location Awareness (NLA)	Nla	Running	Manual
Share Process	c:\windows\system32\svchost.exe -k		
netsvcs	Normal LocalSystem 0		
File Replication	NtFrs	Stopped	Manual
Process	c:\windows\system32\ntfrs.exe	Ignore	Own
LocalSystem 0			
NT LM Security Support Provider	NtLmSsp	Stopped	Manual
Share Process	c:\windows\system32\lsass.exe		
Normal LocalSystem 0			
Removable Storage	NtmsSvc	Stopped	Manual
Process	c:\windows\system32\svchost.exe -k netsvcs		Share
LocalSystem 0			Normal
Plug and Play	PlugPlay	Running	Auto
Share Process	c:\windows\system32\services.exe		Share
LocalSystem 0			Normal
IPSEC Services	PolicyAgent	Running	Auto
Process	c:\windows\system32\lsass.exe		Share
LocalSystem 0			Normal
Protected Storage	ProtectedStorage	Running	Auto
Share Process	c:\windows\system32\lsass.exe		Share
Normal LocalSystem 0			Normal
Remote Access Auto Connection Manager	RasAuto	Stopped	
Manual Share Process			
c:\windows\system32\svchost.exe -k netsvcs			Normal
LocalSystem 0			
Remote Access Connection Manager	RasMan	Stopped	Manual
Share Process	c:\windows\system32\svchost.exe -k		net
netsvcs	Normal LocalSystem 0		
Remote Desktop Help Session Manager	RDSSMgr	Stopped	Manual
Own Process	c:\windows\system32\sessmgr.exe		Normal
LocalSystem 0			
Routing and Remote Access	RemoteAccess	Stopped	
Disabled Share Process			
c:\windows\system32\svchost.exe -k netsvcs			Normal
LocalSystem 0			
Remote Registry	RemoteRegistry	Stopped	Disabled
Share Process	c:\windows\system32\svchost.exe -k		reg
regsvc	Normal NT AUTHORITY\LocalService		0
Remote Procedure Call (RPC) Locator	RpcLocator	Stopped	Manual
Own Process	c:\windows\system32\locator.exe		Normal
NT AUTHORITY\NetworkService 0			
Remote Procedure Call (RPC)	RpcSs	Running	Auto
Share Process	c:\windows\system32\svchost.exe -k		net
rpcss	Normal LocalSystem 0		
Resultant Set of Policy Provider	RSOPProv	Stopped	Manual
Share Process	c:\windows\system32\rsopprov.exe		
Normal LocalSystem 0			
Special Administration Console Helper	sacsvr	Running	Manual
Share Process	c:\windows\system32\svchost.exe -k		net
netsvcs	Normal LocalSystem 0		

Security Accounts Manager	SamSs	Running	Auto	Share
Process	c:\windows\system32\lsass.exe		Normal	
Smart Card	ScardSvr	Stopped	Manual	Share Process
c:\windows\system32\scardsvr.exe			Ignore	NT
AUTHORITY\LocalService 0				
Task Scheduler	Schedule	Running	Auto	Share
Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	
LocalSystem 0				
Secondary Logon	seclogon	Stopped	Disabled	Share
Process	c:\windows\system32\svchost.exe -k netsvcs		Ignore	
LocalSystem 0				
System Event Notification	SENS	Running	Auto	Share
Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	
LocalSystem 0				
Shell Hardware Detection	ShellHWDetection	Running	Auto	
Share Process	c:\windows\system32\svchost.exe -k		net	
netsvcs	Ignore LocalSystem 0			
Print Spooler	Spooler	Stopped	Disabled	Own Process
c:\windows\system32\spoolsv.exe			Normal	
LocalSystem 0				
SQLSERVERAGENT	SQLSERVERAGENT	Stopped	Disabled	
Own Process	c:\program files\microsoft sql			
server\mssql\binn\sqlagent.exe -i mssqlserver			Normal	
LocalSystem 0				
Windows Image Acquisition (WIA)	stisvc	Stopped	Disabled	
Share Process	c:\windows\system32\svchost.exe -k		img	
imgsvc	Normal NT AUTHORITY\LocalService		0	
Microsoft Software Shadow Copy Provider	swprv	Stopped		
Manual Own Process	c:\windows\system32\svchost.exe -k		sw	
swprv	Normal LocalSystem 0			
Performance Logs and Alerts	SysmonLog	Stopped	Manual	
Own Process	c:\windows\system32\smlogsvc.exe		Normal	
NT Authority\NetworkService 0				
Telephony	TapiSrv	Stopped	Manual	Share Process
c:\windows\system32\svchost.exe -k tapisrv			Normal	
LocalSystem 0				
Terminal Services	TermService	Running	Manual	Share
Process	c:\windows\system32\svchost.exe -k term		Normal	
LocalSystem 0				
Telnet	TlntSvr	Stopped	Disabled	Own Process
c:\windows\system32\tlntsvr.exe			Normal	NT
AUTHORITY\LocalService 0				
Distributed Link Tracking Server	TrkSvr	Stopped	Disabled	
Share Process	c:\windows\system32\svchost.exe -k		net	
netsvcs	Normal LocalSystem 0			
Distributed Link Tracking Client	TrkWks	Running	Auto	
Share Process	c:\windows\system32\svchost.exe -k		net	
netsvcs	Normal LocalSystem 0			
Terminal Services Session Directory	Tssdis	Stopped	Disabled	
Own Process	c:\windows\system32\tssdis.exe		Normal	
LocalSystem 0				

Upload Manager	uploadmgr	Stopped	Disabled	Share
Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	
Uninterruptible Power Supply	UPS	Stopped	Manual	
Own Process	c:\windows\system32\ups.exe		Normal	
NT AUTHORITY\LocalService 0				
Virtual Disk Service	vds	Stopped	Manual	Own
Process	c:\windows\system32\vd.exe		Normal	
LocalSystem 0				
Volume Shadow Copy	VSS	Stopped	Manual	Own
Process	c:\windows\system32\vssvc.exe		Normal	
LocalSystem 0				
Windows Time	W32Time	Running	Auto	Share
Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	
LocalSystem 0				
WebClient	WebClient	Stopped	Disabled	Share Process
c:\windows\system32\svchost.exe -k local			Normal	
NT AUTHORITY\LocalService 0				
WinHTTP Web Proxy Auto-Discovery Service	WinHttpAutoProxySvc	Stopped	Manual	Share Process
c:\windows\system32\svchost.exe -k local			Normal	
NT AUTHORITY\LocalService 0				
Windows Management Instrumentation	wimgmt	Running	Auto	
Share Process	c:\windows\system32\svchost.exe -k		net	
netsvcs	Ignore LocalSystem 0			
Windows Management Instrumentation Driver Extensions				Wmi
Stopped Manual Share Process				
c:\windows\system32\svchost.exe -k netsvcs			Normal	
LocalSystem 0				
WMI Performance Adapter	WmiApSrv	Stopped	Manual	Own
Process	c:\windows\system32\wbem\wmiaprv.exe		Normal	
LocalSystem 0				
Automatic Updates	wuauerv	Stopped	Disabled	Share
Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	
LocalSystem 0				
Wireless Configuration	WZCSVC	Stopped	Disabled	Share
Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	
LocalSystem 0				
[Program Groups]				
Group Name	Name	User Name		
Accessories	Default User:Accessories	Default User		
Accessories\Accessibility	Default User:Accessories\Accessibility	Default User		
Accessories\Entertainment	Default User:Accessories\Entertainment	Default User		
Startup	Default User:Startup	Default User		
Accessories	All Users:Accessories	All Users		
Accessories\Accessibility	All Users:Accessories\Accessibility	All Users		
Accessories\Communications	All	All Users		
Users:Accessories\Communications	All Users			

Accessories\Entertainment All Users:Accessories\Entertainment All Users
 Accessories\System Tools All Users:Accessories\System Tools All Users
 Administrative Tools All Users:Administrative Tools All Users
 HP System Tools All Users:HP System Tools All Users
 HP System Tools\hp Array Configuration Utility All Users:HP System
 Tools\hp Array Configuration Utility All Users
 Java Web Start All Users:Java Web Start All Users
 Microsoft SQL Server All Users:Microsoft SQL Server All Users

Startup All Users:Startup All Users
 Accessories NT AUTHORITY\SYSTEM:Accessories NT
 AUTHORITY\SYSTEM
 Accessories\Accessibility NT AUTHORITY\SYSTEM:Accessories\Accessibility
 NT AUTHORITY\SYSTEM
 Accessories\Entertainment NT
 AUTHORITY\SYSTEM:Accessories\Entertainment NT
 AUTHORITY\SYSTEM
 Startup NT AUTHORITY\SYSTEM:Startup NT
 AUTHORITY\SYSTEM
 Accessories SQLOLY\Administrator:Accessories SQLOLY\Administrator

Accessories\Accessibility SQLOLY\Administrator:Accessories\Accessibility
 SQLOLY\Administrator
 Accessories\Entertainment SQLOLY\Administrator:Accessories\Entertainment
 SQLOLY\Administrator
 Administrative Tools SQLOLY\Administrator:Administrative Tools
 SQLOLY\Administrator
 QLogic Corporation SQLOLY\Administrator:QLogic Corporation
 SQLOLY\Administrator
 QLogic Corporation\SANblade Control VIX
 SQLOLY\Administrator:QLogic Corporation\SANblade Control
 VIX SQLOLY\Administrator
 Startup SQLOLY\Administrator:Startup SQLOLY\Administrator

[Startup Programs]

Program	Command	User Name	Location
desktop	desktop.ini	NT AUTHORITY\SYSTEM	Startup
desktop	desktop.ini	SQLOLY\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

Sound Sequence Not Available mplay32.exe /mid
 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	6.0.3790.0
Build	63790
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
actbxprxy.dll	6.0.3790.0	230 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
advpack.dll	6.0.3790.0	240 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
asctrls.ocx	6.0.3790.0	219 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
browsecl.dll	6.0.3790.0	62 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
browseui.dll	6.0.3790.0	2,478 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
cdfview.dll	6.0.3790.0	292 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
comctl32.dll	5.82.3790.0	1,584 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
dxtrans.dll	6.3.3790.0	562 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
dxtmsft.dll	6.3.3790.0	918 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation

iecont.dll	<File Missing>	Not Available	Not Available	Not Available
iecont.dll	<File Missing>	Not Available	Not Available	Not Available
iedkcs32.dll	16.0.3790.0	676 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
iepeers.dll	6.0.3790.0	652 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
iesetup.dll	6.0.3790.0	89 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
ieuinit.inf	Not Available	20 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
ieexplore.exe	6.0.3790.0	102 KB	3/25/2003 4:00:00 AM	C:\Program Files\Internet Explorer
imgutil.dll	5.2.3790.0	101 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
inetctl.cpl	6.0.3790.0	589 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
inetctl.dll	6.0.3790.0	108 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
inseng.dll	6.0.3790.0	213 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
mlang.dll	6.0.3790.0	799 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
msencode.dll	<File Missing>	Not Available	Not Available	Not Available
mshta.exe	6.0.3790.0	59 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
mshtml.dll	6.0.3790.88	8,019 KB	9/22/2003 2:10:36 PM	C:\WINDOWS\system32
mshtml.tlb	6.0.3790.0	1,319 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
mshtml.dll	6.0.3790.0	1,376 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
mshtml.dll	6.0.3790.0	56 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
msident.dll	6.0.3790.0	128 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
msident.dll	6.0.3790.0	14 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
msieft.dll	6.0.3790.0	536 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
msrating.dll	6.0.3790.0	379 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
mstime.dll	6.0.3790.0	1,621 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
occache.dll	6.0.3790.0	201 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32
proctexe.ocx	<File Missing>	Not Available	Not Available	Not Available
sendmail.dll	6.0.3790.0	97 KB	3/25/2003 4:00:00 AM	C:\WINDOWS\system32

```
shdoclc.dll 6.0.3790.0 588 KB 3/25/2003 4:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
shdocvw.dll 6.0.3790.85 3,281 KB 9/22/2003 2:10:36 PM
C:\WINDOWS\system32 Microsoft Corporation
shfolder.dll 6.0.3790.0 37 KB 3/25/2003 4:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
shlwapi.dll 6.0.3790.0 722 KB 3/25/2003 4:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
tdc.ocx 1.3.0.3130 177 KB 3/25/2003 4:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
url.dll 6.0.3790.0 45 KB 3/25/2003 4:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
urlmon.dll 6.0.3790.87 1,242 KB 9/22/2003 2:10:36 PM
C:\WINDOWS\system32 Microsoft Corporation
webcheck.dll 6.0.3790.0 665 KB 3/25/2003 4:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
wininet.dll 6.0.3790.0 1,466 KB 3/25/2003 4:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
```

[Connectivity]

Item	Value
Connection Preference	Never dial

LAN Settings

AutoConfigProxy	Not Available
AutoProxyDetectMode	Disabled
AutoConfigURL	
Proxy	Enabled
ProxyServer	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\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

Update Class Installed
http://v4.windowsupdate.microsoft.com/CAB/ia64/unicode/iuctl.
CAB737910.4515162037
[Content]

[Following are sub-categories of this main category]
[Summary]

Item	Value	Disabled
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 intranet	Medium-low
Trusted sites	Low
Internet	Medium
Restricted sites	High

Microsoft SQL Server 8.0 Startup Parameters

```
sqlservr -c -x -T888 -T825 -T3428 -T3502
```

where
-c : Run as DOS window(Console Mode)
-x: Don't log statistics
T825: enable NUMA support
T888: use locked pages for buffers
-T3502: Display Checkpoint Log to DBLog

-T3428: Speed Up Recovery

Microsoft SQL Server VIA Affinities

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\SuperSocketNetLib\Via]
"Vendor"="QLogic"
"RecognizedVendors"="QLogic"
"ListenInfo"="0:1433[0xF],1:1434[0xF0],2:1435[0xF00],3:1436[0xF000]"
```

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.

SmartArray Driver Settings

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpaqcssb]
"Type"=dword:00000001
"Start"=dword:00000000
"ErrorControl"=dword:00000001
"Tag"=dword:00000102
"ImagePath"=hex(2):73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,44,00,52,00,49,00,56,00,45,00,52,00,53,00,5c,00,68,00,70,00,71,00,63,00,69,00,73,00,73,00,62,00,2e,00,73,00,79,00,73,00,00,00
"DisplayName"="Smart Array Controllers Non-Miniport Bus Driver"
"Group"="port"
```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller0]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller1]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller10]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller11]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller12]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller13]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller14]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller15]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller2]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller3]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller4]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller5]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller6]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller7]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller8]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Parameters\Controller9]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Security]
"Security"=hex:01,00,14,80,90,00,00,00,9c,00,00,00,14,00,00,00,30,00,00,0,0,02,\

00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\

00,00,02,00,60,00,04,00,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,\

05,12,00,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,\

20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,05,0b,00,00,00,0,0,\

00,18,00,fd,01,02,00,01,02,00,00,00,00,00,05,20,00,00,23,02,00,00,01,01,\

00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Enum]

"0"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&28ca0570&0&08"

"Count"=dword:00000010
"NextInstance"=dword:00000010

"1"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&1f34a452&0&08"

"2"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&3858b390&0&08"

"3"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&1e6c4f79&0&08"

"4"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&29a1c50a&0&08"

"5"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&200c63ec&0&08"

"6"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&225f72b7&0&08"
"7"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&fa5f632&0&08"
"8"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&548403b&0&08"
"9"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&edda159&0&08"
"10"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&a466de5&0&08"
"11"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&b10cc7&0&08"
"12"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&19d51c05&0&08"
"13"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&1551cfd&0&08"
"14"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&23cbd7c6&0&08"
"15"="PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\5&2d6138e4&0&08"

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissd]
"Type"=dword:00000001
"Start"=dword:00000000
"ErrorControl"=dword:00000001
"Tag"=dword:00000102
"ImagePath"=hex(2):73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,44,00,\

52,00,49,00,56,00,45,00,52,00,53,00,5c,00,68,00,70,00,71,00,63,00,69,00,73,\

00,73,00,64,00,2e,00,73,00,79,00,73,00,00,00
"DisplayName"="Smart Array Controllers Non-Miniport Disk Driver"
"Group"="Primary Disk"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissd\Security]

"Security"=hex:01,00,14,80,90,00,00,00,9c,00,00,00,14,00,00,00,30,00,00,0,0,02,\

00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\

00,00,02,00,60,00,04,00,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,\

05,12,00,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,\

20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,05,0b,00,00,00,0,0,\
00,18,00,fd,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,02,00,00,01,01,\
00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissd\Enum
]
"0"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&105c4ad
18&0&0000004000000000"
"Count"=dword:00000020
"NextInstance"=dword:00000020
"1"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&105c4ad
18&0&0100004000000000"
"2"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&59190e1
8&0&0000004000000000"
"3"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&59190e1
8&0&0100004000000000"
"4"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&125643b
88&0&0000004000000000"
"5"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&125643b
88&0&0100004000000000"
"6"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&3a74778
8&0&0000004000000000"
"7"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&3a74778
8&0&0100004000000000"
"8"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&188e934
8&0&0000004000000000"
"9"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&188e934
8&0&0100004000000000"
"10"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&2cbb64
81&0&0000004000000000"
"11"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&2cbb64
81&0&0100004000000000"
"12"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&15c777
69&0&0000004000000000"
"13"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&15c777
69&0&0100004000000000"
"14"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&4f9649b
8&0&0000004000000000"
"15"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&4f9649b
8&0&0100004000000000"
"16"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&325b62
44&0&0000004000000000"
"17"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&325b62
44&0&0100004000000000"
"18"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&4a2ae8
d8&0&0000004000000000"
"19"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&4a2ae8
d8&0&0100004000000000"
```

```
"20"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&23248c
e9&0&0000004000000000"
"22"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&d419c7
e&0&0000004000000000"
"23"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&d419c7
e&0&0100004000000000"
"24"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&16bc63
46&0&0000004000000000"
"25"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&16bc63
46&0&0100004000000000"
"26"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&b6267c
98&0&0000004000000000"
"27"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&b6267c
98&0&0100004000000000"
"28"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&cf9a53c
8&0&0000004000000000"
"29"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&cf9a53c
8&0&0100004000000000"
"30"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&265ec5
22&0&0000004000000000"
"31"="HPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\6&265ec5
22&0&0100004000000000"
```

Qlogic 2350 VIA Card Settings

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2300]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000000
"Tag"=dword:00000028
"Type"=dword:00000001
"ImagePath"=hex(2):73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,
,44,00,\  
52,00,49,00,56,00,45,00,52,00,53,00,5c,00,71,00,6c,00,32,00,33,00,30,00,30,
,00,2e,00,73,00,79,00,73,00,00,00,\  
"DisplayName"="ql2300"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2300\Parame
ters]
"BusType"=dword:00000006
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2300\Parame
ters\Device]
"CreateInitiatorLU"=dword:00000001
"DriverParameters"="UseSameNN=0"
"MaximumSGList"=dword:00000021
"NumberOfRequests"=dword:00000096
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2300\Parame
ters\PnpInterface]
"5"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2300\Enum]
"0"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&9c102bf&
0&08"
"Count"=dword:00000004
"NextInstance"=dword:00000004
"1"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&37d3486a
&0&08"
"2"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&245142ec
&0&08"
"3"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&3666e35b
&0&08"
```

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qlvika]
"start"=dword:00000002
"type"=dword:00000001
"Tag"=dword:00000001
"group"="MVIA"
"ErrorControl"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qlvika\Adapter
s]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qlvika\Adapter
s\210000E08B0ECFCD]
"IPAddress"=hex(7):31,00,35,00,2e,00,32,00,2e,00,31,00,30,00,30,00,2e,00,
32,00,\  
32,00,00,00,00,00,00,\  
"VIDeviceNumber"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qlvika\Adapter
s\210000E08B0ED1CD]
"IPAddress"=hex(7):31,00,35,00,2e,00,32,00,2e,00,31,00,30,00,30,00,2e,00,
32,00,\  
31,00,00,00,00,00,00,\  
"VIDeviceNumber"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qlvika\Adapter
s\210000E08B0ED2CD]
```

```
[HKKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qlvika\Enum]
"0"="Root\\LEGACY_QLVIKA\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

name	minimum	maximum	config_value	run_value
affinity mask	-2147483648	2147483647	65535	
65535				
affinity64 mask	-2147483648	2147483647	0	0
allow updates	0	1	0	0
awe enabled	0	1	0	0
c2 audit mode	0	1	0	0
cost threshold for parallelism	0	32767	5	5
Cross DB Ownership Chaining		0	1	0
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
max server memory (MB)		4	2147483647	124000
124000				
max text repl size (B)	0	2147483647	65536	65536

media retention	0	365	0	0
max worker threads (KB)	32512	2147483647	850	102450
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	0	0
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

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP]
"NOTE"="This is for backward compatibility only."
```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters]

"MajorVersion"=dword:00000005
"MinorVersion"=dword:00000000
"InstallPath"="C:\\WINNT\\System32\\inetrv"
"CertMapList"="C:\\WINNT\\System32\\inetrv\\iisrmap.dll"
"AccessDeniedMessage"="Error: Access is Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\\WINNT\\System32\\LogFiles"
"AcceptExOutstanding"=dword:00000028

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots]

"/"="c:\\inetpub\\wwwroot,,205"
"/Scripts"="c:\\inetpub\\scripts,,1"
"/IISHelp"="c:\\winnt\\help\\iishelp,,1"
"/IISAdmin"="C:\\WINNT\\System32\\inetrv\\iisadmin,,1"
"/IISSamples"="c:\\inetpub\\iissamples,,1"
"/MSADC"="c:\\program files\\common files\\system\\msadc,,1"
"/Printers"="C:\\WINNT\\web\\printers,,201"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Performance]

"Library"="w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"WbemAdapFileSize"=hex:00,d8,60,11,b3,1,c2,01
"WbemAdapFileSize"=dword:00001d10
"WbemAdapStatus"=dword:00000000
"Last Counter"=dword:00000c64
"Last Help"=dword:00000c65
"First Counter"=dword:00000bc2
"First Help"=dword:00000bc3
"Library Validation
Code"=hex:4e,7b,1d,b9,0b,f9,c2,01,10,3d,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\

00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00,00,\

00,00,02,00,70,00,04,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,\

05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,05,\

20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\

00,00,05,0b,00,00,00,20,02,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\

00,05,20,00,00,23,02,00,00,72,00,73,00,01,01,00,00,00,00,05,12,00,00,\

00,01,01,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Enum]
"0"="Root\\LEGACY_W3SVC\\0000"

"Count"=dword:00000001
"NextInstance"=dword:00000001

TPCC Application Registry Parameters

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPCC]

"NumberOfDeliveryThreads"=dword:0000000a
"MaxConnections"=dword:000055f0
"MaxPendingDeliveries"=dword:000007d0
"DB_Protocol"="ODBC"
"TxnMonitor"="COM"
"Path"="C:\\Inetpub\\wwwroot\\"
"DbServer"="via:sqloly"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"="ssdl"
"COM_SinglePool"="YES"
"SPPrefix"="sp3"

TCP/IP Registry Parameters

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters]

"NV Hostname"="w32"
"DataBasePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,\

00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,\

64,00,72,00,69,00,76,00,65,00,72,00,73,00,5c,00,65,00,74,00,63,00,00,00

"NameServer"=""

"ForwardBroadcasts"=dword:00000000

"IPEnableRouter"=dword:00000000

"Domain"=""

"Hostname"="w32"

"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"="15.1.101.1"

"DhcpDomain"="hp-perf.net"

Microsoft Windows 2000 Server Configuration Parameters

System Information report written at: 12/10/2003 09:17:25 AM
[System Information]

[Following are sub-categories of this main category]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Server
Version	5.0.2195 Service Pack 3 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	W32

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 W32\Administrator
Time Zone Pacific Standard Time
Total Physical Memory 1,048,084 KB
Available Physical Memory 852,212 KB
Total Virtual Memory 4,094,440 KB
Available Virtual Memory 3,775,264 KB
Page File Space 3,046,356 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

Address Range	Device	Status
0x1800-0x18FF	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	Keyboard	OK
0x0064-0x0064	Standard 101/102-Key or Microsoft Natural PS/2	
Keyboard	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

Address Range	Device	Status
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 Si
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
14	Primary IDE Channel
7	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 Si	OK
0xF5DF0000-0xF5DF3FFF	Compaq Smart Array Si	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
0xF5E80000-0xF5EFFFFFF	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\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\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\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\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					
c:\winnt\system32\tssoft32.acm	DSP GROUP, INC.		OK	C:\WINNT\System32\TSOFT32.ACM	1.01
9.27 KB (9,488 bytes)	12/31/1979 4:00:00 PM				
c:\winnt\system32\iac25_32.ax	Intel Corporation		Indeo® audio software	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\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					

[Video Codecs]

Codec	Manufacturer	Description	Status	File	Version
Size	Creation Date				
c:\winnt\system32\ir50_32.dll	Intel Corporation		Indeo® video 5.10	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\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\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\iccvld.dll	Radius Inc.	OK
C:\WINNT\System32\ICCVLD.DLL	1.10.0.6	108.00
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\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\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

[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 Rate	Not 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 Type	ATI RAGE XL PCI, ATI Technologies Inc. compatible
Adapter Description	ATI Technologies Inc. RAGE XL PCI
Adapter RAM	8.00 MB (8,388,608 bytes)
Installed Drivers	atidrab.dll
Driver Version	5.00.2179.1
INF File	display.inf (atirage3 section)
Color Planes	1
Color Table Entries	65536
Resolution	640 x 480 x 60 hertz
Bits/Pixel	16

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 Type	Ethernet 802.3
Product Name	Compaq NC7781 Gigabit Server Adapter
Installed	True
PNP Device ID	PCI\VEN_14E4&DEV_16A7&SUBSYS_00C80E11&REV_02\3&10700208&0&10
Last Reset	12/8/2003 3:27:18 AM
Index	0
Service Name	q57w2k
IP Address	15.1.101.50

IP Subnet 255.255.0.0
Default IP Gateway 15.1.101.1
DHCP Enabled True
DHCP Server 15.1.101.1
DHCP Lease Expires 12/16/2003 11:27:48 AM
DHCP Lease Obtained 12/8/2003 11:27:48 AM
MAC Address 00:0B:CD:50:44:10
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 Type Ethernet 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 12/8/2003 3:27:18 AM
Index 1
Service Name q57w2k
IP Address
IP Subnet
Default IP Gateway Not Available
DHCP Enabled True
DHCP Server
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:0B:CD:50:44:2D
Service Name q57w2k
IRQ Number 30
Driver c:\winnt\system32\drivers\q57w2k.sys (77776, 2.75.0.0)

Name [00000002] RAS Async Adapter
Adapter Type Not Available
Product Name RAS Async Adapter
Installed True
PNP Device ID Not Available
Last Reset 12/8/2003 3:27:18 AM
Index 2
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Not Available

Name [00000003] WAN Miniport (L2TP)

Product Name Not Available WAN Miniport (L2TP)
Installed True
PNP Device ID ROOT\MS_L2TPMINIPOINT\0000
Last Reset 12/8/2003 3:27:18 AM
Index 3
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Rasl2tp
Driver c:\winnt\system32\drivers\rasl2tp.sys (52112, 5.00.2195.4052)

Name [00000004] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Name WAN Miniport (PPTP)
Installed True
PNP Device ID ROOT\MS_PPTPMINIPOINT\0000
Last Reset 12/8/2003 3:27:18 AM
Index 4
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30
Service Name PptpMiniport
Driver c:\winnt\system32\drivers\raspptp.sys (47888, 5.00.2195.4080)

Name [00000005] Direct Parallel
Adapter Type Not Available
Product Name Direct Parallel
Installed True
PNP Device ID ROOT\MS_PTMINIPOINT\0000
Last Reset 12/8/2003 3:27:18 AM
Index 5
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Raspti

Driver c:\winnt\system32\drivers\raspti.sys (16880, 5.00.2146.1)
Name [00000006] WAN Miniport (IP)
Adapter Type Not Available
Product Name WAN Miniport (IP)
Installed True
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 12/8/2003 3:27:18 AM
Index 6
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name NdisWan
Driver c:\winnt\system32\drivers\ndiswan.sys (93104, 5.00.2195.5241)

[Protocol]

Item	Value
Name	MSAFD Tcpip [TCP/IP]
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	16 bytes
MaximumMessageSize	0 bytes
MessageOriented	False
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	True
SupportsGracefulClosing	True
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD Tcpip [UDP/IP]
ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	16 bytes
MaximumMessageSize	65467 bytes
MessageOriented	True
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True

SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	True

Name	RSVP UDP Service Provider
ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	16 bytes
MaximumMessageSize	65467 bytes
MessageOriented	True
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	True
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	True

Name	RSVP TCP Service Provider
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	16 bytes
MaximumMessageSize	0 bytes
MessageOriented	False
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	True
SupportsExpeditedData	True
SupportsGracefulClosing	True
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{E098C77C-5847-452B-9532-634577082B5F}] SEQUENCE 0
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True

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}] SEQUENCE 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

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}] SEQUENCE 2
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	20 bytes
MaximumMessageSize	64000 bytes
MessageOriented	True
MinimumAddressSize	20 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{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}] SEQUENCE 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\wsock32.dll
Version	5.00.2195.4874
Size	21.27 KB (21,776 bytes)

[Ports]

[Following are sub-categories of this main category]

[Serial]

Item	Value
Name	COM1
Status	OK
PNP Device ID	ACPI\PNP0501\0
Maximum Input Buffer Size	0
Maximum Output Buffer Size	False
Settable Baud Rate	True
Settable Data Bits	True
Settable Flow Control	True
Settable Parity	True
Settable Parity Check	True
Settable Stop Bits	True
Settable RLSD	True
Supports RLSD	True
Supports 16 Bit Mode	False
Supports Special Characters	False
Baud Rate	9600
Bits/Byte	8
Stop Bits	1
Parity	None
Busy	0
Abort Read/Write on Error	0
Binary Mode Enabled	-1
Continue XMit on XOff	0
CTS Outflow Control	0
Discard NULL Bytes	0
DSR Outflow Control	0
DSR Sensitivity	0
DTR Flow Control Type	Enable
EOF Character	0
Error Replace Character	0
Error Replacement Enabled	0
Event Character	0
Parity Check Enabled	0
RTS Flow Control Type	Enable
XOff Character	19
XOffXMit Threshold	512
XOn Character	17
XOnXMit Threshold	2048
XOnXOff InFlow Control	0
XOnXOff OutFlow Control	0
IRQ Number	4
I/O Port	0x03F8-0x03FF
Driver	c:\winnt\system32\drivers\serial.sys (62512, 5.00.2195.5080)

[Parallel]

Item	Value
No parallel port information	

[Storage]

[Following are sub-categories of this main category]

[Drives]

Item	Value
Drive	A:
Description	3 1/2 Inch Floppy Drive
Drive	C:
Description	Local Fixed Disk
Compressed	False
File System	NTFS
Size	33.91 GB (36,406,377,984 bytes)
Free Space	28.59 GB (30,701,623,808 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 SCSIbus	0
Drive SCSILogicalUnit	0
Drive SCSIPort	2
Drive SCSTargetId	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&267
A616A0&20	
Device ID	PCI\VEN_0E11&DEV_B178&SUBSYS_40800E11&REV_01\3&267
A616A0&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
 Device Map 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&07A
 USB Root Hub USB\ROOT_HUB\4&AF5358C&0

[Software Environment]

[Following are sub-categories of this main category]

[Drivers]

Name	Description	File	Type	Started	Start
Mode	State	Status	Error Control	Accept	Pause Accept
Stop					

abiosdsk	Abiosdsk	Not Available	Kernel Driver	False	Disabled
abp480n5	abp480n5	Not Available	Kernel Driver	False	Disabled
acpi	Microsoft ACPI Driver	c:\winnt\system32\drivers\acpi.sys	Kernel Driver	True	Running
acpiec	ACPIEC	c:\winnt\system32\drivers\acpiec.sys	Kernel Driver	False	Stopped
adpu160m	adpu160m	c:\winnt\system32\drivers\adpu160m.sys	Kernel Driver	True	Running
afd	AFD Networking Support Environment	c:\winnt\system32\drivers\afd.sys	Kernel Driver	True	Running
aha154x	Aha154x	Not Available	Kernel Driver	False	Disabled
aic116x	aic116x	Not Available	Kernel Driver	False	Disabled
aic78u2	aic78u2	Not Available	Kernel Driver	False	Disabled
aic78xx	aic78xx	Not Available	Kernel Driver	False	Disabled
ami0nt	ami0nt	Not Available	Kernel Driver	False	Disabled
amsint	amsint	Not Available	Kernel Driver	False	Disabled
asc	asc	Not Available	Kernel Driver	False	Disabled
asc3350p	asc3350p	Not Available	Kernel Driver	False	Disabled
asc3550	asc3550	Not Available	Kernel Driver	False	Disabled
asynmac	RAS Asynchronous Media Driver	c:\winnt\system32\drivers\asynmac.sys	Kernel Driver	False	Stopped
atapi	Standard IDE/ESDI Hard Disk Controller	c:\winnt\system32\drivers\atapi.sys	Kernel Driver	True	Running
atdisk	Atdisk	Not Available	Kernel Driver	False	Disabled
atirage3	atirage3	c:\winnt\system32\drivers\atimpab.sys	Kernel Driver	True	Running
atmarpc	ATM ARP Client Protocol	c:\winnt\system32\drivers\atmarpc.sys	Kernel Driver	False	Stopped

audstub	Audio Stub Driver	c:\winnt\system32\drivers\audstub.sys	Kernel Driver	True	Running
beep	Beep	c:\winnt\system32\drivers\beep.sys	Kernel Driver	True	Running
buslogic	BusLogic	Not Available	Kernel Driver	False	Disabled
cd20xrnt	cd20xrnt	Not Available	Kernel Driver	False	Disabled
cdaudio	Cdaudio	c:\winnt\system32\drivers\cdaudio.sys	Kernel Driver	False	Stopped
cdfs	Cdfs	c:\winnt\system32\drivers\cdfs.sys	File System Driver	True	Running
cdrom	CD-ROM Driver	c:\winnt\system32\drivers\cdrom.sys	Kernel Driver	True	Running
changer	Changer	Not Available	Kernel Driver	False	System
cpq32fs2	cpq32fs2	c:\winnt\system32\drivers\cpq32fs2.sys	Kernel Driver	True	Running
cpqarray	Cpqarray	c:\winnt\system32\drivers\cpqarray.sys	Kernel Driver	True	Running
cpqarry2	cpqarry2	c:\winnt\system32\drivers\cpqarry2.sys	Kernel Driver	True	Running
cpqasm2	hp ProLiant iLO Advanced System Management Controller	c:\winnt\system32\drivers\cpqasm2.sys	Kernel Driver	True	Running
cpqcdrv	HP Integrated Lights-Out	c:\winnt\system32\drivers\cpqcdrv.sys	Kernel Driver	True	Running
cpqcisse	CPQCISSE	c:\winnt\system32\drivers\cpqcisse.sys	Kernel Driver	True	Running
cpqcissm	cpqcissm	c:\winnt\system32\drivers\cpqcissm.sys	Kernel Driver	True	Running
cpqfcalm	cpqfcalm	Not Available	Kernel Driver	False	Disabled
cpqfws2e	cpqfws2e	Not Available	Kernel Driver	False	Disabled
cpqteam	Compaq Network Teaming and Configuration	c:\winnt\system32\drivers\cpqteam.sys	Kernel Driver	False	Stopped

dac960nt	dac960nt	Not Available	Kernel Driver	False	Disabled
	Stopped	OK	Normal	False	False
deckzpsx	deckzpsx	Not Available	Kernel Driver	False	Disabled
	Stopped	OK	Normal	False	False
dfsdriver	DfsDriver	c:\winnt\system32\drivers\dfs.sys	File		
System Driver	True	Boot	Running	OK	
	Normal	False	True		
disk	Disk Driver	c:\winnt\system32\drivers\disk.sys	Kernel		
Driver	True	Boot	Running	OK	Normal
	False	True			
diskperf	Diskperf	c:\winnt\system32\drivers\diskperf.sys	Kernel		
Driver	True	Boot	Running	OK	Normal
	False	True			
dmboot	dmboot	c:\winnt\system32\drivers\dmboot.sys	Kernel		
Driver	False	Disabled	Stopped	OK	Normal
	False	False			
dmio	Logical Disk Manager Driver	c:\winnt\system32\drivers\dmio.sys	Kernel Driver	True	
	Boot	Running	OK	Normal	False
	True				
dmload	dmload	c:\winnt\system32\drivers\dmload.sys	Kernel		
Driver	True	Boot	Running	OK	Normal
	False	True			
efs	EFS	c:\winnt\system32\drivers\efs.sys	File		
System Driver	True	Disabled	Running	OK	
	Normal	False	True		
fastfat	Fastfat	c:\winnt\system32\drivers\fastfat.sys	File		
System Driver	True	Disabled	Running	OK	
	Normal	False	True		
fd16_700	Fd16_700	Not Available	Kernel Driver	False	Disabled
	Stopped	OK	Normal	False	False
fdc	Floppy Disk Controller Driver	c:\winnt\system32\drivers\fdc.sys	Kernel Driver	True	
	Manual	Running	OK	Normal	False
	True				
fips	Fips	c:\winnt\system32\drivers\fips.sys	Kernel		
Driver	True	Auto	Running	OK	Normal
	False	True			
fireport	fireport	Not Available	Kernel Driver	False	Disabled
	Stopped	OK	Normal	False	False
flashpnt	flashpnt	Not Available	Kernel Driver	False	Disabled
	Stopped	OK	Normal	False	False
flpydisk	Floppy Disk Driver	c:\winnt\system32\drivers\flpydisk.sys	Kernel Driver	True	
	Manual	Running	OK	Normal	False
	True				
ftdisk	Volume Manager Driver	c:\winnt\system32\drivers\ftdisk.sys	Kernel Driver	True	
	Kernel Driver	True	Boot	Running	OK
	Normal	False	True		

gpc	Generic Packet Classifier	c:\winnt\system32\drivers\msgpc.sys	Kernel Driver	True	
	Manual	Running	OK	Normal	False
hidusb	Microsoft HID Class Driver	c:\winnt\system32\drivers\hidusb.sys	Kernel Driver	False	
	Auto	Stopped	OK	Ignore	False
	False				
i8042prt	i8042 Keyboard and PS/2 Mouse Port Driver	c:\winnt\system32\drivers\i8042prt.sys	Kernel Driver	True	
	System	Running	OK	Normal	False
	True				
ini910u	ini910u	Not Available	Kernel Driver	False	Disabled
	Stopped	OK	Normal	False	False
intellide	IntelIde	Not Available	Kernel Driver	False	Disabled
	Stopped	OK	Normal	False	False
ipfilterdriver	IP Traffic Filter Driver	c:\winnt\system32\drivers\ipfltdrv.sys	Kernel Driver	False	
	Manual	Stopped	OK	Normal	False
	False				
ipinip	IP in IP Tunnel Driver	c:\winnt\system32\drivers\ipinip.sys	Kernel Driver	False	
	Kernel Driver	Manual	Stopped	OK	
	Normal	False	False		
ipnat	IP Network Address Translator	c:\winnt\system32\drivers\ipnat.sys	Kernel Driver	False	
	Manual	Stopped	OK	Normal	False
	False				
ipsec	IPSEC driver	c:\winnt\system32\drivers\ipsec.sys	Kernel		
Driver	True	Manual	Running	OK	Normal
	False	True			
ipsraidn	ipsraidn	Not Available	Kernel Driver	False	Disabled
	Stopped	OK	Normal	False	False
irenum	IR Enumerator Service	c:\winnt\system32\drivers\irenum.sys	Kernel Driver	False	
	Manual	Stopped	OK	Normal	False
	False				
isapnp	PnP ISA/EISA Bus Driver	c:\winnt\system32\drivers\isapnp.sys	Kernel Driver	True	
	Boot	Running	OK	Critical	False
	True				
kbdclass	Keyboard Class Driver	c:\winnt\system32\drivers\kbdclass.sys	Kernel Driver	True	
	System	Running	OK	Normal	False
	True				
kbdhid	Keyboard HID Driver	c:\winnt\system32\drivers\kbdhid.sys	Kernel Driver	False	
	System	Stopped	OK	Ignore	False
	False				
ksecdd	KSecDD	c:\winnt\system32\drivers\ksecdd.sys	Kernel		
Driver	True	Boot	Running	OK	Normal
	False	True			
lbrtdc	lbrtdc	Not Available	Kernel Driver	False	System
	Stopped	OK	Ignore	False	False

Ip6nds35	Ip6nds35	Not Available	Kernel Driver	False	Disabled
megaide	megaide	Not Available	Kernel Driver	False	Disabled
Driver	Stopped	OK	Normal	False	Normal
	False	False			
mnmd	mnmd	c:\winnt\system32\drivers\mnmd.sys	Kernel		
Driver	True	System	Running	OK	Ignore
	False	True			
modem	Modem	c:\winnt\system32\drivers\modem.sys	Kernel		
Driver	False	Manual	Stopped	OK	Ignore
	False	False			
mouclass	Mouse Class Driver	c:\winnt\system32\drivers\mouclass.sys	Kernel Driver	True	
	System	Running	OK	Normal	False
	True				
mouhid	Mouse HID Driver	c:\winnt\system32\drivers\mouhid.sys	Kernel Driver	False	
	Manual	Stopped	OK	Ignore	False
	False				
mountmgr	MountMgr	c:\winnt\system32\drivers\mountmgr.sys	Kernel Driver	True	
	Normal	False	Boot	Running	OK
	False	True			
mrraid35x	mrraid35x	Not Available	Kernel Driver	False	Disabled
	Stopped	OK	Normal	False	False
mrxsmb	MRXSMB	c:\winnt\system32\drivers\mrxsmb.sys	File		
System Driver	True	System	Running	OK	
	Normal	False	True		
msfs	Msfs	c:\winnt\system32\drivers\msfs.sys	File		
System Driver	True	System	Running	OK	
	Normal	False	True		
mskssrv	Microsoft Streaming Service Proxy	c:\winnt\system32\drivers\mskssrv.sys	Kernel Driver	False	
	Manual	Stopped	OK	Normal	False
	False				
mspclock	Microsoft Streaming Clock Proxy	c:\winnt\system32\drivers\mspclock.sys	Kernel Driver	False	
	Manual	Stopped	OK	Normal	False
	False				
mspqm	Microsoft Streaming Quality Manager Proxy	c:\winnt\system32\drivers\mspqm.sys	Kernel Driver	False	
	Manual	Stopped	OK	Normal	False
	False				
mup	Mup	c:\winnt\system32\drivers\mup.sys	File		
System Driver	True	Boot	Running	OK	
	Normal	False	True		
ncrc710	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	
	Kernel Driver	True	Boot	Running	OK
	Normal	False	True		
ndistapi	Remote Access NDIS TAPI Driver	c:\winnt\system32\drivers\ndistapi.sys	Kernel Driver	True	
	Manual	Running	OK	Normal	False
	True				

ndiswan	Remote Access NDIS WAN Driver	c:\winnt\system32\drivers\ndiswan.sys	Kernel Driver	True	True	Manual	Running	OK	Normal	False
ndproxy	NDIS Proxy	c:\winnt\system32\drivers\ndproxy.sys	Kernel Driver	True	True	Manual	Running	OK	Normal	False
netbios	NetBIOS Interface	c:\winnt\system32\drivers\netbios.sys	File System Driver	True	True	System	Running	OK	Normal	False
netbt	NetBios over Tcpip	c:\winnt\system32\drivers\netbt.sys	Kernel Driver	True	True	System	Running	OK	Normal	False
netdetect	NetDetect	c:\winnt\system32\drivers\netdetect.sys	Kernel Driver	True	True	Manual	Stopped	OK	Normal	False
npfs	Npfs	c:\winnt\system32\drivers\npfs.sys	File System Driver	True	True	System	Running	OK	Normal	False
ntfs	Ntfs	c:\winnt\system32\drivers\ntfs.sys	File System Driver	True	True	Disabled	Running	OK	Normal	False
null	Null	c:\winnt\system32\drivers\null.sys	Kernel Driver	True	True	System	Running	OK	Normal	False
nwlnkflt	IPX Traffic Filter Driver	c:\winnt\system32\drivers\nwlnkflt.sys	Kernel Driver	True	True	Manual	Stopped	OK	Normal	False
nwlnkfw	IPX Traffic Forwarder Driver	c:\winnt\system32\drivers\nwlnkfw.sys	Kernel Driver	True	True	Manual	Stopped	OK	Normal	False
openhci	Microsoft USB Open Host Controller Driver	c:\winnt\system32\drivers\openhci.sys	Kernel Driver	True	True	Manual	Running	OK	Normal	False
parallel	Parallel	c:\winnt\system32\drivers\parallel.sys	Kernel Driver	True	True	Manual	Stopped	OK	Ignore	False
parport	Parport	c:\winnt\system32\drivers\parport.sys	Kernel Driver	True	True	Manual	Stopped	OK	Ignore	False
partmgr	PartMgr	c:\winnt\system32\drivers\partmgr.sys	Kernel Driver	True	True	Boot	Running	OK	Normal	False
parvdm	ParVdm	c:\winnt\system32\drivers\parvdm.sys	Kernel Driver	True	True	Manual	Stopped	OK	Ignore	False

pci	PCI Bus Driver	c:\winnt\system32\drivers\pci.sys	Kernel Driver	True	True	Boot	Running	OK	Normal	False
pcidump	PCI Dump	c:\winnt\system32\drivers\pcidump.sys	Kernel Driver	True	True	Manual	Stopped	OK	Ignore	False
pciide	PCIIDE	c:\winnt\system32\drivers\pciide.sys	Kernel Driver	True	True	Boot	Running	OK	Normal	False
pcmcia	Pcmcia	c:\winnt\system32\drivers\pcmcia.sys	Kernel Driver	True	True	Disabled	Stopped	OK	Normal	False
pdcomp	PDCOMP	c:\winnt\system32\drivers\pdcomp.sys	Kernel Driver	True	True	Not Available	Stopped	OK	Ignore	False
pdframe	PDFRAME	c:\winnt\system32\drivers\pdframe.sys	Kernel Driver	True	True	Not Available	Stopped	OK	Ignore	False
pdreli	PDRELI	c:\winnt\system32\drivers\pdreli.sys	Kernel Driver	True	True	Not Available	Stopped	OK	Ignore	False
pdrframe	PDRFRAME	c:\winnt\system32\drivers\pdrframe.sys	Kernel Driver	True	True	Not Available	Stopped	OK	Ignore	False
pptpminiport	WAN Miniport (PPTP)	c:\winnt\system32\drivers\rasppptp.sys	Kernel Driver	True	True	Manual	Running	OK	Normal	False
ptilink	Direct Parallel Link Driver	c:\winnt\system32\drivers\ptilink.sys	Kernel Driver	True	True	Manual	Running	OK	Normal	False
q57w2k	Compaq NC7781 Gigabit Server Adapter	c:\winnt\system32\drivers\q57w2k.sys	Kernel Driver	True	True	Manual	Running	OK	Normal	False
ql1080	ql1080	c:\winnt\system32\drivers\ql1080.sys	Kernel Driver	True	True	Not Available	Stopped	OK	Normal	False
ql10wnt	ql10wnt	c:\winnt\system32\drivers\ql10wnt.sys	Kernel Driver	True	True	Not Available	Stopped	OK	Normal	False
ql1240	ql1240	c:\winnt\system32\drivers\ql1240.sys	Kernel Driver	True	True	Not Available	Stopped	OK	Normal	False
ql2100	ql2100	c:\winnt\system32\drivers\ql2100.sys	Kernel Driver	True	True	Not Available	Stopped	OK	Normal	False
ql2300	ql2300	c:\winnt\system32\drivers\ql2300.sys	Kernel Driver	True	True	Boot	Running	OK	Normal	False
qlvika	qlvika	c:\winnt\system32\drivers\qlvika.sys	Kernel Driver	True	True	Auto	Running	OK	Normal	False
rasacd	Remote Access Auto Connection Driver	c:\winnt\system32\drivers\rasacd.sys	Kernel Driver	True	True	System	Running	OK	Normal	False
rasl2tp	WAN Miniport (L2TP)	c:\winnt\system32\drivers\rasl2tp.sys	Kernel Driver	True	True	Manual	Running	OK	Normal	False

raspti	Direct Parallel	c:\winnt\system32\drivers\raspti.sys	Kernel Driver	True	True	Manual	Running	OK	Normal	False
rca	Microsoft Streaming Network Raw Channel Access	c:\winnt\system32\drivers\rca.sys	Kernel Driver	True	True	Manual	Stopped	OK	Normal	False
rdbss	Rdbss	c:\winnt\system32\drivers\rdbss.sys	File System Driver	True	True	System	Running	OK	Normal	False
rdpdr	Terminal Server Device Redirector Driver	c:\winnt\system32\drivers\rdpdr.sys	Kernel Driver	True	True	Manual	Running	OK	Normal	False
rdpwd	RDPWD	c:\winnt\system32\drivers\rdpwd.sys	Kernel Driver	True	True	Manual	Running	OK	Ignore	False
redbook	Digital CD Audio Playback Filter Driver	c:\winnt\system32\drivers\redbook.sys	Kernel Driver	True	True	System	Stopped	OK	Normal	False
serenum	Serenum Filter Driver	c:\winnt\system32\drivers\serenum.sys	Kernel Driver	True	True	Manual	Running	OK	Normal	False
serial	Serial port driver	c:\winnt\system32\drivers\serial.sys	Kernel Driver	True	True	Manual	Running	OK	Normal	False
sfloppy	Sfloppy	c:\winnt\system32\drivers\sfloppy.sys	Kernel Driver	True	True	System	Stopped	OK	Ignore	False
sglfb	sglfb	c:\winnt\system32\drivers\sglfb.sys	Kernel Driver	True	True	Not Available	Stopped	OK	Normal	False
simbad	Simbad	c:\winnt\system32\drivers\simbad.sys	Kernel Driver	True	True	Not Available	Stopped	OK	Normal	False
sparrow	Sparrow	c:\winnt\system32\drivers\sparrow.sys	Kernel Driver	True	True	Not Available	Stopped	OK	Normal	False
spud	Special Purpose Utility Driver	c:\winnt\system32\drivers\spud.sys	Kernel Driver	True	True	Manual	Running	OK	Normal	False
srv	Srv	c:\winnt\system32\drivers\srv.sys	File System Driver	True	True	Manual	Running	OK	Normal	False
swenum	Software Bus Driver	c:\winnt\system32\drivers\swenum.sys	Kernel Driver	True	True	Manual	Running	OK	Normal	False
symc810	symc810	c:\winnt\system32\drivers\symc810.sys	Kernel Driver	True	True	Boot	Running	OK	Normal	False
symc8xx	symc8xx	c:\winnt\system32\drivers\symc8xx.sys	Kernel Driver	True	True	Boot	Running	OK	Normal	False

sym_hi Driver	sym_hi True	c:\winnt\system32\drivers\sym_hi.sys	Kernel Boot Running OK	Normal
sysmgmt	hp ProLiant System Management Interface Driver	c:\winnt\system32\drivers\sysmgmt.sys	Kernel Driver Manual Running OK	True Normal False
tcpip	TCP/IP Protocol Driver	c:\winnt\system32\drivers\tcpip.sys	Kernel Driver Normal True	System Running OK
tdasync Driver	TDASYNC	c:\winnt\system32\drivers\tdasync.sys	Kernel False Manual Stopped	OK Ignore
tdipx Driver	TDIPX	c:\winnt\system32\drivers\tdipx.sys	Kernel False Manual Stopped	OK Ignore
tdnetb Driver	TDNETB	c:\winnt\system32\drivers\tdnetb.sys	Kernel False Manual Stopped	OK Ignore
tdpipe Driver	TDPIPE	c:\winnt\system32\drivers\tdpipe.sys	Kernel False Manual Stopped	OK Ignore
tdspix Driver	TDSPX	c:\winnt\system32\drivers\tdspix.sys	Kernel False Manual Stopped	OK Ignore
tdtcp Driver	TDTCP	c:\winnt\system32\drivers\tdtcp.sys	Kernel True Manual Running	OK Ignore
termdd	Terminal Device Driver	c:\winnt\system32\drivers\termdd.sys	Kernel Driver Auto Running	True Normal False
tga	tga	Not Available	Kernel Driver Stopped OK	False Ignore False
udfs System Driver	Udfs	c:\winnt\system32\drivers\udfs.sys	File False Disabled Stopped	OK
ultra66	ultra66	Not Available	Kernel Driver Stopped OK	False Disabled False
update	Microcode Update Driver	c:\winnt\system32\drivers\update.sys	Kernel Driver Manual Running	True Normal False
usbhub	Microsoft USB Standard Hub Driver	c:\winnt\system32\drivers\usbhub.sys	Kernel Driver Manual Running	True Normal False
vgaSave Driver	VgaSave	c:\winnt\system32\drivers\vga.sys	Kernel True System Running	OK Ignore

wanarp	Remote Access IP ARP Driver	c:\winnt\system32\drivers\wanarp.sys	Kernel Driver Manual Running	True Normal False
wdica	WDICA	Not Available	Kernel Driver Stopped OK	False Ignore False

[Environment Variables]

Variable	Value	User Name
ComSpec	%SystemRoot%\system32\cmd.exe	<SYSTEM>
Os2LibPath	%SystemRoot%\system32\os2\dll;	<SYSTEM>
Path	%SystemRoot%\system32;%SystemRoot%\%SystemRoot%\System32\Wbem;C:\Program Files\Microsoft SQL Server\80\Tools\BINN	<SYSTEM>
windir	%SystemRoot%	<SYSTEM>
OS	Windows_NT	<SYSTEM>
PROCESSOR_ARCHITECTURE	x86	<SYSTEM>
PROCESSOR_LEVEL	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	W32\Administrator
TMP	%USERPROFILE%\Local Settings\Temp	W32\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	
Processor	Job ID	Priority	Parameters	Driver Name	Print
No print jobs	Host Print Queue	Data Type	Name		

[Network Connections]

Local Name	Remote Name	Type	Status	User
E:	\\ssdlib\d\$	Disk	OK	

[Running Tasks]

Name	Path	Process ID	Priority	Min Working Set	Version	Size
system idle process	Not Available	0	0	0	0	Not
Available	Not Available	Not Available	Unknown	Unknown	Unknown	Unknown
system	Not Available	8	8	0	0	1413120
smss.exe	c:\winnt\system32\smss.exe	204800	1413120	12/8/2003 11:27:40 AM	236	11
	5.00.2195.5382		44.77 KB (45,840 bytes)			
	12/31/1979 4:00:00 PM					
csrss.exe	Not Available	260	13	Not Available	Not	Not
Available	12/8/2003 11:27:45 AM	Unknown	Unknown	Unknown	Unknown	Unknown
winlogon.exe	c:\winnt\system32\winlogon.exe	204800	1413120	12/8/2003 11:27:46 AM	256	13
	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	12/8/2003 11:27:46 AM	312	9
	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	12/8/2003 11:27:47 AM	324	9
	5.00.2195.5430		32.77 KB (33,552 bytes)			
	12/31/1979 4:00:00 PM					
termsrv.exe	c:\winnt\system32\termsrv.exe	204800	1413120	12/8/2003 11:27:47 AM	436	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	12/8/2003 11:27:48 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	12/8/2003 11:27:48 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	12/8/2003 11:27:48 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	12/8/2003 11:27:49 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	12/8/2003 11:27:49 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	12/8/2003 11:27:49 AM	812	9
	5.00.2195.4907		81.27 KB (83,216 bytes)			
	7/22/2002 1:05:04 PM					

```

sqlservr.exe c:\progra~1\micro~3\mssql\bin\sqlservr.exe 840
8 204800 1413120 12/8/2003 11:27:49
AM 2000.080.0194.00 7.10 MB (7,442,493 bytes)
12/4/2003 10:08:18 AM
regsvc.exe c:\winnt\system32\regsvc.exe 908 8
204800 1413120 12/8/2003 11:27:50 AM
5.00.2195.3649 65.27 KB (66,832 bytes) 4/2/2003
1:16:42 PM
rsys.exe Not Available 928 8 Not Available Not
Available 12/8/2003 11:27:50 AM Unknown Unknown Unknown
mstask.exe c:\winnt\system32\mstask.exe 940 8
204800 1413120 12/8/2003 11:27:50 AM
4.71.2195.1 115.77 KB (118,544 bytes) 4/2/2003 1:16:37 PM
snmp.exe c:\winnt\system32\snmp.exe 992 8
204800 1413120 12/8/2003 11:27:50 AM
5.00.2195.5080 29.77 KB (30,480 bytes) 4/2/2003
3:33:32 AM
winmgmt.exec c:\winnt\system32\wbem\winmgmt.exe 1068 8
204800 1413120 12/8/2003 11:27:51 AM
1.50.1085.0070 192.08 KB (196,685 bytes) 4/2/2003
1:16:50 PM
svchost.exe c:\winnt\system32\svchost.exe 1088 8
204800 1413120 12/8/2003 11:27:51 AM
5.00.2134.1 7.77 KB (7,952 bytes) 12/31/1979 4:00:00
PM
dfssvc.exe c:\winnt\system32\dfssvc.exe 1108 8
204800 1413120 12/8/2003 11:27:51 AM
5.00.2195.3649 88.27 KB (90,384 bytes) 4/2/2003
1:16:28 PM
mssearch.exe c:\program files\common
files\system\mssearch\bin\mssearch.exe 1144 8 204800
1413120 12/8/2003 11:27:51 AM 9.107.5512.0 72.00 KB
(73,728 bytes) 2/7/2003 3:54:48 PM
sysdown.exe c:\winnt\system32\sysdown.exe 1240 8
204800 1413120 12/8/2003 11:27:53 AM
5.26.2195.0 34.77 KB (35,600 bytes) 4/2/2003 11:42:10 AM
svchost.exe c:\winnt\system32\svchost.exe 1628 8
204800 1413120 12/8/2003 11:28:04 AM
5.00.2134.1 7.77 KB (7,952 bytes) 12/31/1979 4:00:00
PM
logon.scr c:\winnt\system32\logon.scr 652 4
204800 1413120 12/8/2003 11:42:51 AM
5.00.2195.5305 127.77 KB (130,832 bytes) 4/2/2003
1:16:33 PM
dllhost.exe Not Available 1728 8 Not Available Not
Available 12/8/2003 2:17:24 PM Unknown Unknown Unknown
inetinfo.exe c:\winnt\system32\inetrv\inetinfo.exe 1780 8
204800 1413120 12/8/2003 4:25:01 PM
5.00.0984 14.27 KB (14,608 bytes) 4/2/2003 1:17:01 PM
csrss.exe Not Available 1820 13 Not Available Not
Available 12/8/2003 4:58:00 PM Unknown Unknown Unknown

```

```

winlogon.exe c:\winnt\system32\winlogon.exe 1872 13
204800 1413120 12/8/2003 4:58:00 PM
5.00.2195.5386 174.77 KB (178,960 bytes) 4/2/2003
1:16:40 PM
rdpclip.exe c:\winnt\system32\rdpclip.exe 7984 8
204800 1413120 12/8/2003 4:58:02 PM
5.00.2174.1 39.77 KB (40,720 bytes) 4/2/2003 3:33:34 AM
explorer.exe c:\winnt\explorer.exe 7720 8 204800
1413120 12/8/2003 4:58:02 PM 5.00.3502.5321
237.27 KB (242,960 bytes) 4/2/2003 1:16:47 PM
cpqteam.exe c:\winnt\system32\cpqteam.exe 7628 8
204800 1413120 12/8/2003 4:58:02 PM
7.10.710.18 84.09 KB (86,112 bytes) 6/12/2002 8:58:50 AM
sqlmangr.exe c:\program files\microsoft sql
server\80\tools\bin\sqlmangr.exe 7856 8 204800
1413120 12/8/2003 4:58:03 PM 2000.080.0194.00
68.00 KB (69,632 bytes) 12/4/2003 10:08:43 AM
wuauclt.exe c:\winnt\system32\wuauclt.exe 7804 8
204800 1413120 12/8/2003 4:58:17 PM
5.4.3628.1 built by: lab04_n 137.50 KB (140,800
bytes) 4/2/2003 1:17:00 PM
cmd.exe c:\winnt\system32\cmd.exe 7908 8
204800 1413120 12/10/2003 9:06:18 AM
5.00.2195.4803 230.77 KB (236,304 bytes)
12/31/1979 4:00:00 PM
mmc.exe c:\winnt\system32\mmc.exe 1544 8
204800 1413120 12/10/2003 9:16:38 AM
5.00.2195.4933 589.27 KB (603,408 bytes) 4/2/2003
1:16:33 PM
rsvp.exe c:\winnt\system32\rsvp.exe 7564 8
204800 1413120 12/10/2003 9:17:10 AM
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 Microsoft Corporation c:\winnt\system32\traffic.dll
rsvp.exe 5.00.2167.1 172.77 KB (176,912 bytes) 12/31/1979 4:00:00
PM Microsoft Corporation c:\winnt\system32\rsvp.exe
wbemprox.dll 1.50.1085.0045 40.08 KB (41,040 bytes) 4/2/2003
1:16:50 PM Microsoft Corporation
c:\winnt\system32\wbem\wbemprox.dll
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
mmcnmgr.dll 5.00.2195.5352 816.27 KB (835,856
bytes) 4/2/2003 1:16:33 PM Microsoft Corporation
c:\winnt\system32\mmcnmgr.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

```

```

cmd.exe 5.00.2195.4803 230.77 KB (236,304 bytes)
12/31/1979 4:00:00 PM Microsoft Corporation
riched20.dll 5.00.2195.4803 230.77 KB (236,304 bytes)
4/2/2003 1:16:42 PM Microsoft Corporation
c:\winnt\system32\riched20.dll
wuauclt.exe 5.4.3628.1 built by: lab04_n 137.50 KB (140,800
bytes) 4/2/2003 1:17:00 PM Microsoft Corporation
c:\winnt\system32\wuauclt.exe
sqlmangr.rll 2000.080.0194.00 96.00 KB (98,304 bytes)
12/4/2003 10:08:43 AM Microsoft Corporation
c:\program files\microsoft sql
server\80\tools\bin\resources\1033\sqlmangr.rll
sqlsvc.rll 2000.080.0194.00 24.00 KB (24,576 bytes)
12/4/2003 10:08:41 AM Microsoft Corporation
c:\program files\microsoft sql
server\80\tools\bin\resources\1033\sqlsvc.rll
sqlresld.dll 2000.080.0194.00 28.06 KB (28,738 bytes)
12/4/2003 10:08:41 AM Microsoft Corporation
c:\program files\microsoft sql server\80\tools\bin\sqlresld.dll
odbcdbc.dll 2000.081.9031.010 24.00 KB (24,576 bytes)
10/22/2002 5:20:20 PM Microsoft Corporation
c:\winnt\system32\odbcdbc.dll
sqlsvc.dll 2000.080.0194.00 92.06 KB (94,272 bytes)
12/4/2003 10:08:41 AM Microsoft Corporation
c:\program files\microsoft sql server\80\tools\bin\sqlsvc.dll
sqlunirl.dll 2000.080.0728.00 176.56 KB (180,800 bytes)
10/2/2002 5:32:28 PM Microsoft Corporation
c:\winnt\system32\sqlunirl.dll
w95scm.dll 2000.080.0194.00 48.06 KB (49,216 bytes)
12/4/2003 10:08:41 AM Microsoft Corporation
c:\program files\microsoft sql server\80\tools\bin\w95scm.dll
sqlmangr.exe 2000.080.0194.00 68.00 KB (69,632 bytes)
12/4/2003 10:08:43 AM Microsoft Corporation
c:\program files\microsoft sql
server\80\tools\bin\sqlmangr.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
imm32.dll 5.00.2195.4314 93.77 KB (96,016 bytes) 4/2/2003
1:16:31 PM Microsoft Corporation c:\winnt\system32\imm32.dll
msdbg.dll 6.00.8424 67.50 KB (69,120 bytes) 4/2/2003 3:34:25 AM
Microsoft Corporation c:\winnt\system32\msdbg.dll
pdm.dll 6.00.8424 179.27 KB (183,574 bytes) 4/2/2003 3:34:25 AM
Microsoft Corporation c:\winnt\system32\pdm.dll
mlang.dll 5.00.3315.3727 509.77 KB (522,000 bytes) 4/2/2003
1:16:33 PM Microsoft Corporation c:\winnt\system32\mlang.dll
shdocl.dll 5.00.3502.5039 324.50 KB (332,288 bytes) 4/2/2003
1:16:43 PM Microsoft Corporation c:\winnt\system32\shdocl.dll
urlmon.dll 5.00.3502.5400 442.27 KB (452,880 bytes) 4/2/2003
1:16:45 PM Microsoft Corporation c:\winnt\system32\urlmon.dll
browser.dll 5.00.3502.4373 34.50 KB (35,328 bytes) 4/2/2003
1:16:25 PM Microsoft Corporation c:\winnt\system32\browser.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		
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		
cscui.dll	5.00.2195.4104	233.77 KB (239,376 bytes)	4/2/2003 1:16:27 PM
Microsoft Corporation	c:\winnt\system32\cscui.dll		
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\inetrv\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\inetrv\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\inetrv\md5filt.dll		
seo.dll	5.00.0984 229.27 KB (234,768 bytes)	4/2/2003 1:17:02 PM	
Microsoft Corporation	c:\winnt\system32\inetrv\seo.dll		
gzip.dll	5.00.0984 30.27 KB (30,992 bytes)	4/2/2003 1:17:01 PM	
Microsoft Corporation	c:\winnt\system32\inetrv\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\inetrv\compfilt.dll		

sspicfilt.dll	5.00.0984 42.77 KB (43,792 bytes)	4/2/2003 1:17:02 PM	
Microsoft Corporation	c:\winnt\system32\inetrv\sspicfilt.dll		
iscomlog.dll	5.00.0984 24.27 KB (24,816 bytes)	4/2/2003 1:17:02 PM	
Microsoft Corporation	c:\winnt\system32\inetrv\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\inetrv\lonsint.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		
smtpsvc.dll	5.00.0984 428.27 KB (438,544 bytes)	4/2/2003 1:17:02 PM	
Microsoft Corporation	c:\winnt\system32\inetrv\smtpsvc.dll		
inetsloc.dll	5.00.0984 20.27 KB (20,752 bytes)	4/2/2003 1:16:31 PM	
Microsoft Corporation	c:\winnt\system32\inetsloc.dll		
iisfecnv.dll	5.00.0984 7.27 KB (7,440 bytes)	4/2/2003 3:33:45 AM	
Microsoft Corporation	c:\winnt\system32\inetrv\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\inetrv\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\inetrv\infocomm.dll		
w3svc.dll	5.00.0984 335.27 KB (343,312 bytes)	4/2/2003 1:17:02 PM	
Microsoft Corporation	c:\winnt\system32\inetrv\w3svc.dll		
svcxext.dll	5.00.0984 39.77 KB (40,720 bytes)	4/2/2003 1:17:02 PM	
Microsoft Corporation	c:\winnt\system32\inetrv\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\inetrv\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\inetrv\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\inetrv\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\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\inetrv\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\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\inetrv\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\inetrv\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\inetrv\rpcref.dll		
inetinfo.exe	5.00.0984 14.27 KB (14,608 bytes)	4/2/2003 1:17:02 PM	
Microsoft Corporation	c:\winnt\system32\inetrv\inetinfo.exe		
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		
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		
iprop.dll	5.00.2195.5023	4.27 KB (4,368 bytes)	4/2/2003 1:16:32 PM
Microsoft Corporation	c:\winnt\system32\iprop.dll		
srchidx.dll	9.107.5512.0 433.50 KB (443,904 bytes)	2/7/2003 3:54:52 PM	
Microsoft Corporation	c:\program~1\common~1\system\mssearch\bin\srchidx.dll		
propdefs.dll	9.107.5512.0 164.00 KB (167,936 bytes)	2/7/2003 3:54:51 PM	
Microsoft Corporation	c:\program~1\common~1\system\mssearch\bin\propdefs.dll		
tquery.dll	9.107.5512.0 1.61 MB (1,690,112 bytes)	2/7/2003 3:54:52 PM	
Microsoft Corporation	c:\program files\common files\system\mssearch\bin\tquery.dll		
mssrch.dll	9.107.5512.0 1.49 MB (1,566,976 bytes)	2/7/2003 3:54:49 PM	
Microsoft Corporation	c:\program~1\common~1\system\mssearch\bin\mssrch.dll		
mssws.dll	9.107.5512.0 18.94 KB (19,392 bytes)	2/7/2003 3:54:50 PM	
Microsoft Corporation	c:\program files\common files\system\mssearch\bin\mssws.dll		
mssearch.exe	9.107.5512.0 72.00 KB (73,728 bytes)	2/7/2003 3:54:48 PM	
Microsoft Corporation	c:\program files\common files\system\mssearch\bin\mssearch.exe		
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		
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		
utildll.dll	5.00.2153.1 25.77 KB (26,384 bytes)	12/31/1979 4:00:00 PM	
Microsoft Corporation	c:\winnt\system32\utildll.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 built by: lab04_n	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 built by: lab04_n	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		
wbemsvcl.dll	1.50.1085.0007	40.07 KB (41,036 bytes)	4/2/2003 1:16:50 PM
Microsoft Corporation	c:\winnt\system32\wbem\wbemsvcl.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		
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		
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		
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		
winnmgmt.exe	1.50.1085.0070	192.08 KB (196,685 bytes)	4/2/2003 1:16:50 PM
Microsoft Corporation	c:\winnt\system32\wbem\winnmgmt.exe		
odbcint.dll	3.520.9001.0 88.00 KB (90,112 bytes)	4/2/2003 1:33:04 PM	
Microsoft Corporation	c:\winnt\system32\odbcint.dll		
odbc32.dll	3.520.9030.0 196.00 KB (200,704 bytes)	4/2/2003 1:33:04 PM	
Microsoft Corporation	c:\winnt\system32\odbc32.dll		
sqlsnmp.dll	2000.080.0194.00	60.00 KB (61,440 bytes)	12/4/2003 10:08:19 AM
Microsoft Corporation	c:\program files\microsoft sql server\mssql\bin\sqlsnmp.dll		
iisrt.dll	5.00.0984 119.77 KB (122,640 bytes)	4/2/2003 1:16:31 PM	
Microsoft Corporation	c:\winnt\system32\iisrt.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\httpmib.dll		
rtipxmib.dll	5.00.2168.1 13.27 KB (13,584 bytes)	12/31/1979 4:00:00 PM	
Microsoft Corporation	c:\winnt\system32\rtipxmib.dll		
btapagnt.dll	5.00.2168.1 13.27 KB (13,584 bytes)	12/31/1979 4:00:00 PM	
Microsoft Corporation	c:\winnt\system32\btapagnt.dll		
ospfagnt.dll	5.00.2168.1 6.77 KB (6,928 bytes)	12/31/1979 4:00:00 PM	
Microsoft Corporation	c:\winnt\system32\ospfagnt.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		
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		
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		
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		
lmib2.dll	5.00.2195.4874	29.27 KB (29,968 bytes)	4/2/2003 1:16:33 PM
Microsoft Corporation	c:\winnt\system32\lmib2.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		
oledb32r.dll	2.70.9001.0 built by: Lab06_N(dagbuild)	64.00 KB (65,536 bytes)	4/2/2003 1:33:04 PM
Microsoft Corporation	c:\program files\common files\system\ole db\oledb32r.dll		
oledb32.dll	2.71.9031.4 built by: Lab06_N(dagbuild)	408.00 KB (417,792 bytes)	9/27/2002 1:22:42 PM
Microsoft Corporation	c:\program files\common files\system\ole db\oledb32.dll		
msdatl.dll	2.71.9030.0 built by: Lab06_N(dagbuild)	84.00 KB (86,016 bytes)	4/2/2003 1:33:04 PM
Microsoft Corporation	c:\program files\common files\system\ole db\msdatl.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		

msdart.dll	2.71.9031.4 built by: Lab06_N(dagbuild)	124.00 KB (126,976 bytes)	9/27/2002 1:22:40 PM
Microsoft Corporation	c:\program files\microsoft sql server\msdart.dll		
sqloledb.dll	2000.080.0194.00	460.00 KB (471,040 bytes)	10/9/2002 10:57:42 AM
Microsoft Corporation	c:\program files\common files\system\ole db\sqloledb.dll		
sqlftqry.dll	2000.080.0194.00	108.07 KB (110,668 bytes)	12/4/2003 10:08:26 AM
Microsoft Corporation	c:\progra~1\micro~3\mssql\bin\sqlftqry.dll		
ssmslpcn.dll	2000.080.0194.00	28.06 KB (28,734 bytes)	12/4/2003 10:08:19 AM
Microsoft Corporation	c:\progra~1\micro~3\mssql\bin\ssmslpcn.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		
ssnmpn70.dll	2000.080.0194.00	24.06 KB (24,638 bytes)	12/4/2003 10:08:19 AM
Microsoft Corporation	c:\progra~1\micro~3\mssql\bin\ssnmpn70.dll		
ssnetlib.dll	2000.080.0194.00	84.06 KB (86,078 bytes)	12/4/2003 10:08:19 AM
Microsoft Corporation	c:\progra~1\micro~3\mssql\bin\ssnetlib.dll		
sqllevn70.rll	2000.080.0194.00	28.00 KB (28,672 bytes)	12/4/2003 10:08:20 AM
Microsoft Corporation	c:\progra~1\micro~3\mssql\bin\resources\1033\sqllevn70.rll		
msvcirt.dll	6.10.8637.0 76.05 KB (77,878 bytes)	12/31/1979 4:00:00 PM	
Microsoft Corporation	c:\winnt\system32\msvcirt.dll		
sqlsort.dll	2000.080.0194.00	576.06 KB (589,885 bytes)	12/4/2003 10:08:19 AM
Microsoft Corporation	c:\progra~1\micro~3\mssql\bin\sqlsort.dll		
ums.dll	2000.080.0194.00	48.06 KB (49,210 bytes)	12/4/2003 10:08:19 AM
Microsoft Corporation	c:\progra~1\micro~3\mssql\bin\ums.dll		
opends60.dll	2000.080.0194.00	24.06 KB (24,639 bytes)	12/4/2003 10:08:19 AM
Microsoft Corporation	c:\progra~1\micro~3\mssql\bin\opends60.dll		
sqlservr.exe	2000.080.0194.00	7.10 MB (7,442,493 bytes)	12/4/2003 10:08:18 AM
Microsoft Corporation	c:\progra~1\micro~3\mssql\bin\sqlservr.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		
llsrv.exe	5.00.2195.4907	81.27 KB (83,216 bytes)	7/22/2002 1:05:04 PM
Microsoft Corporation	c:\winnt\system32\llsrv.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
1:16:38 PM	Microsoft Corporation	c:\winnt\system32\netcfgx.dll
rasmans.dll	5.00.2195.5436	149.27 KB (152,848 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
msvcps50.dll	5.00.7051 552.50 KB (565,760 bytes)	12/31/1979 4:00:00
PM	Microsoft Corporation	c:\winnt\system32\msvcps50.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
pjlmon.dll	5.00.2165.1 12.77 KB (13,072 bytes)	11/30/1999 3:39:36
PM	Microsoft Corporation	c:\winnt\system32\pjlmon.dll
cnbjmon.dll	5.00.2134.1 43.77 KB (44,816 bytes)	11/30/1999 3:38:48
PM	Microsoft Corporation	c:\winnt\system32\cnbjmon.dll

localspl.dll	5.00.2195.5423	250.27 KB (256,272 bytes)
	12/31/1979 4:00:00 PM	Microsoft Corporation
	c:\winnt\system32\localspl.dll	
spoolss.dll	5.00.2195.4299	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
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
iisuba.dll	5.00.0984 9.77 KB (10,000 bytes)	12/31/1979 4:00:00
PM	Microsoft Corporation	c:\winnt\system32\iisuba.dll
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
ntdsa.dll	5.00.2195.5438	1002.27 KB (1,026,320 bytes)
	4/2/2003 1:16:39 PM	Microsoft Corporation
	c:\winnt\system32\ntdsa.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	5.00.2195.5201	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
1:16:38 PM	Microsoft Corporation	c:\winnt\system32\netlogon.dll
kerberos.dll	5.00.2195.5246	202.77 KB (207,632 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	
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	
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
winmr.dll	5.00.2160.1 18.77 KB (19,216 bytes)	12/31/1979 4:00:00
PM	Microsoft Corporation	c:\winnt\system32\winmr.dll
rnrd20.dll	5.00.2195.4874	35.77 KB (36,624 bytes) 4/2/2003
1:16:42 PM	Microsoft Corporation	c:\winnt\system32\rnrd20.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
msgsvc.dll	5.00.2195.4874	34.77 KB (35,600 bytes)
	12/31/1979 4:00:00 PM	Microsoft Corporation
	c:\winnt\system32\msgsvc.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\srsvsc.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 1:16:47 PM	Microsoft Corporation
c:\winnt\system32\wshtcpip.dll				
msafd.dll	5.00.2195.4874	103.27 KB (105,744 bytes)	4/2/2003 1:16:34 PM	Microsoft Corporation
c:\winnt\system32\msafd.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				
umpnpgm.dll	5.00.2182.1	86.27 KB (88,336 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\umpnpgm.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	5.00.2134.1	114.960 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				
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				
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				
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				

wsock32.dll	5.00.2195.4874	21.27 KB (21,776 bytes)	4/2/2003 1:16:28 PM	Microsoft Corporation
c:\winnt\system32\wsock32.dll				
dnsapi.dll	5.00.2134.1	17.77 KB (18,192 bytes)	12/31/1979 4:00:00 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 1:16:47 PM	Microsoft Corporation
c:\winnt\system32\ws2_32.dll				
samlib.dll	5.00.2195.4827	49.77 KB (50,960 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\samlib.dll				
netrap.dll	5.00.2134.1	11.27 KB (11,536 bytes)	12/31/1979 4:00:00 PM	Microsoft Corporation
c:\winnt\system32\netrap.dll				
netapi32.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 Type
Alerter	c:\winnt\system32\services.exe	Running	Auto	Share Process
Application Management Process	c:\winnt\system32\services.exe	Stopped	Manual	Share Normal
Background Intelligent Transfer Service	c:\winnt\system32\svchost.exe -k	Running	Auto	Share Process
bitsgroup	LocalSystem 0	Running	Auto	Share Process
Computer Browser	c:\winnt\system32\services.exe	Running	Auto	Share Process
Indexing Service	c:\winnt\system32\cisvc.exe	Stopped	Manual	Share Process
ClipBook	c:\winnt\system32\clipsrv.exe	Stopped	Manual	Own Process
Compaq Remote Monitor Service	c:\winnt\system32\cpqrcmc.exe	Running	Auto	Share Normal
Distributed File System	c:\winnt\system32\dfsrv.exe	Running	Auto	Own Process
DHCP Client	c:\winnt\system32\services.exe	Running	Auto	Share Process
Logical Disk Manager	c:\winnt\system32\dmadmin.exe /com	Running	Manual	Share Process
Logical Disk Manager	c:\winnt\system32\dmserver.exe	Running	Auto	Share Process
DNS Client	c:\winnt\system32\services.exe	Running	Auto	Share Process
Event Log	c:\winnt\system32\services.exe	Running	Auto	Share Process
COM+ Event System	c:\winnt\system32\svchost.exe -k netsvcs	Running	Manual	Share Normal
Fax Service	c:\winnt\system32\faxsvc.exe	Stopped	Manual	Own Process

IIS Admin Service	IISADMIN	Running	Auto	Share Process
InterSite Messaging	c:\winnt\system32\ismsserv.exe	Stopped	Disabled	Own Process
Kerberos Key Distribution Center	c:\winnt\system32\lsass.exe	Running	Auto	Share Process
Server	c:\winnt\system32\services.exe	Running	Auto	Share Process
Workstation	c:\winnt\system32\services.exe	Running	Auto	Share Process
License Logging Service	c:\winnt\system32\llssrv.exe	Running	Auto	Share Process
TCP/IP NetBIOS Helper Service	c:\winnt\system32\services.exe	Running	Auto	Share Process
Messenger	c:\winnt\system32\services.exe	Running	Auto	Share Process
NetMeeting Remote Desktop Sharing	c:\winnt\system32\mnmsrvc.exe	Stopped	Manual	Share Normal
Distributed Transaction Coordinator	c:\winnt\system32\msdtc.exe	Running	Auto	Share Normal
Windows Installer	c:\winnt\system32\msiexec.exe /v	Running	Manual	Share Process
Microsoft Search	c:\program files\common files\system\mssearch\bin\mssearch.exe	Running	Auto	Share Process
MSSQLSERVER	c:\program files\microsoft sql server\tools\bin\sqladhip.exe	Running	Auto	Share Process
MSSQLServerADHelper	c:\program files\microsoft sql server\tools\bin\sqladhip.exe	Running	Auto	Share Process
Network DDENetDDE	c:\winnt\system32\netdde.exe	Running	Auto	Share Process
Network DDE DSDM	c:\winnt\system32\netdde.exe	Running	Auto	Share Process
Net Logon	c:\winnt\system32\lsass.exe	Running	Auto	Share Process

Network Connections	Netman	Running	Manual	Share Process
File Replication Service	c:\winnt\system32\svchost.exe -k netsvcs	Running	Manual	Share Process
NTFS	c:\winnt\system32\ntfs.sys	Running	Manual	Share Process
NT LM Security Support Provider	c:\winnt\system32\lsass.exe	Running	Manual	Share Process
Removable Storage	c:\winnt\system32\svchost.exe -k netsvcs	Running	Auto	Share Process
Plug and Play	c:\winnt\system32\services.exe	Running	Auto	Share Process
IPSEC Policy Agent	c:\winnt\system32\ipsec.sys	Running	Auto	Share Process
Protected Storage	c:\winnt\system32\services.exe	Running	Auto	Share Process
Remote Access Auto Connection Manager	c:\winnt\system32\svchost.exe -k netsvcs	Running	Auto	Share Process
Remote Access Connection Manager	c:\winnt\system32\svchost.exe -k netsvcs	Running	Auto	Share Process
Routing and Remote Access	c:\winnt\system32\svchost.exe -k netsvcs	Running	Auto	Share Process
Remote Registry Service	c:\winnt\system32\regsvcs.exe	Running	Auto	Share Process
Remote Command Service	c:\program files\benchmark\rsys.exe	Running	Auto	Share Process
Remote Procedure Call (RPC) Locator	c:\winnt\system32\locator.exe	Running	Auto	Share Process
Remote Procedure Call (RPC)	c:\winnt\system32\svchost.exe -k rpcss	Running	Auto	Share Process
QoS RSVP	c:\winnt\system32\rsvp.exe -s	Running	Auto	Share Process
Security Accounts Manager	c:\winnt\system32\lsass.exe	Running	Auto	Share Process
Smart Card Helper	c:\winnt\system32\scardsvr.exe	Running	Auto	Share Process

Smart Card	SCardSvr	Stopped	Manual	Share Process
	c:\winnt\system32\scardsvr.exe			Ignore
Task Scheduler		Schedule	Running	Auto
Process	c:\winnt\system32\mtask.exe			Share
	LocalSystem	0		Normal
RunAs Service	seclogon	Running	Auto	Share
Process	c:\winnt\system32\services.exe			Ignore
	LocalSystem	0		
System Event Notification	SENS	Running	Auto	Share
Process	c:\winnt\system32\svchost.exe -k netsvcs			Normal
	LocalSystem	0		
Internet Connection Sharing		SharedAccess		Stopped
	Manual	Share Process		
	c:\winnt\system32\svchost.exe -k netsvcs			Normal
	LocalSystem	0		
Simple Mail Transport Protocol (SMTP)	SMTPSVC	Running	Auto	
	Share Process			
	c:\winnt\system32\inetrv\inetinfo.exe			Normal
	LocalSystem	0		
SNMP Service	SNMP	Running	Auto	Own
Process	c:\winnt\system32\snmp.exe			Normal
	LocalSystem	0		
SNMP Trap Service	SNMPTRAP	Stopped	Manual	Own
Process	c:\winnt\system32\snmptrap.exe			Normal
	LocalSystem	0		
Print Spooler	Spooler	Running	Auto	Own Process
	c:\winnt\system32\spoolsv.exe			Normal
	LocalSystem	0		
SQLSERVERAGENT	SQLSERVERAGENT	Stopped	Manual	
	Own Process	c:\progra~1\micro~3\mssql\binn\sqlagent.exe		
	Normal	LocalSystem	0	
hp ProLiant System Shutdown Service	sysdown	Running	Auto	
	Own Process	c:\winnt\system32\sysdown.exe		Normal
	LocalSystem	0		
Performance Logs and Alerts	SysmonLog	Stopped	Manual	
	Own Process	c:\winnt\system32\smlogsvc.exe		Normal
	LocalSystem	0		
Telephony	TapiSrv	Running	Manual	Share Process
	c:\winnt\system32\svchost.exe -k tapisrv			Normal
	LocalSystem	0		
Terminal Services	TermService	Running	Auto	Own
Process	c:\winnt\system32\termsrv.exe			Normal
	LocalSystem	0		
Telnet	TlntSvr	Stopped	Manual	Own Process
	c:\winnt\system32\tlntsvr.exe			Normal
	LocalSystem	0		
Distributed Link Tracking Server	TrkSvr	Stopped	Manual	
	Share Process	c:\winnt\system32\services.exe		
	Normal	LocalSystem	0	

Distributed Link Tracking Client	TrkWks	Running	Auto
	Share Process	c:\winnt\system32\services.exe	
Uninterruptible Power Supply	UPS	Stopped	Manual
	Own Process	c:\winnt\system32\ups.exe	Normal
	LocalSystem	0	
Utility Manager	UtilMan	Stopped	Manual
Process	c:\winnt\system32\utilman.exe		Own
	LocalSystem	0	Normal
Windows Time	W32Time	Stopped	Manual
Process	c:\winnt\system32\services.exe		Share
	LocalSystem	0	Normal
World Wide Web Publishing Service	W3SVC	Running	Auto
	Share Process		
	c:\winnt\system32\inetrv\inetinfo.exe		Normal
	LocalSystem	0	
Windows Management Instrumentation	WinMgmt	Running	Auto
	Own Process	c:\winnt\system32\wbem\winmgmt.exe	Ignore
	LocalSystem	0	
Windows Management Instrumentation Driver Extensions			Wmi
	Running	Manual	Share Process
	c:\winnt\system32\services.exe		Normal
	LocalSystem	0	
Automatic Updates	wuauclt	Running	Auto
Process	c:\winnt\system32\svchost.exe -k wugroup		Share
	LocalSystem	0	Normal

[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	
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
Microsoft SQL Server - Switch	All Users:Microsoft SQL Server - Switch	All Users
Startup	All Users:Startup	All Users
Accessories	W32\Administrator:Accessories	W32\Administrator

Accessories\Accessibility	W32\Administrator:Accessories\Accessibility
Accessories\Entertainment	W32\Administrator:Accessories\Entertainment
Accessories\System Tools	W32\Administrator:Accessories\System Tools
Benchcraft	W32\Administrator:Benchcraft
QLogic Corporation	W32\Administrator:QLogic Corporation
QLogic Corporation\SANblade Control VIX	W32\Administrator:QLogic Corporation\SANblade Control VIX
Startup	W32\Administrator:Startup

[Startup Programs]

Program	Command	User Name	Location
Microsoft Office	c:\progra~1\micro~4\office\osa9.exe -b -l	All Users	Common Startup
CPQTEAM	cpqteam.exe	All Users	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

[OLE Registration]

Object	Local Server
Microsoft Excel Worksheet	Not Available
Microsoft Excel Chart	Not Available
Microsoft Excel Worksheet	C:\PROGRA~1\MICROS~4\Office\EXCEL.EXE
Microsoft Excel Chart	C:\PROGRA~1\MICROS~4\Office\EXCEL.EXE
Sound (OLE2)	sndrec32.exe
Media Clip	mplay32.exe
Video Clip	mplay32.exe /avi
MIDI Sequence	mplay32.exe /mid
	Not Available
Microsoft Excel Chart	Not Available
Microsoft Excel 4.0 Macro	Not Available
Sound	Not Available
Media Clip	Not Available
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		
PM	C:\WINNT\system32	Microsoft Corporation			
advpack.dll	5.0.3502.4373	87 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
browsecl.dll	5.0.3502.4373	35 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
browseui.dll	5.0.3502.4373	791 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
ckcnv.exe	5.0.2189.1 9 KB	12/7/1999 4:00:00 AM			
PM	C:\WINNT\system32	Microsoft Corporation			
comctl32.dll	5.81.3315.3727	539 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
crypt32.dll	5.131.2195.4558	464 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
enhsg.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available
Available	Not Available				
iemigrat.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available
Available	Not Available				
iesetup.dll	5.0.3502.4373	57 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
ieexplore.exe	5.0.2920.0 59 KB	12/7/1999 4:00:00 AM			
PM	C:\Program Files\Internet Explorer	Microsoft Corporation			
imagehlp.dll	5.0.2195.5242	126 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
imghelp.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available
Available	Not Available				
inseng.dll	5.0.3502.4373	72 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
jobexec.dll	5.0.0.1 47 KB	12/7/1999 4:00:00 AM			
PM	C:\WINNT\system32	Microsoft Corporation			
jscrip.dll	5.6.0.6626 576 KB	6/26/2001 6:06:52 PM			
PM	C:\WINNT\system32	Microsoft Corporation			
jsproxy.dll	5.0.2920.0 13 KB	12/7/1999 4:00:00 AM			
PM	C:\WINNT\system32	Microsoft Corporation			
msaahtml.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available
Available	Not Available				
mshhtml.dll	5.0.3502.5390	2284 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
msjava.dll	5.0.3805.0 924 KB	7/22/2002 12:05:04 PM			
PM	C:\WINNT\system32	Microsoft Corporation			

msoss.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available
Available	Not Available				
ole32.dll	5.0.2195.5400	86 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
oleaut32.dll	5.0.2195.5400	968 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
olepro32.dll	5.0.4518.0 612 KB	7/22/2002 12:05:04 PM			
PM	C:\WINNT\system32	Microsoft Corporation			
olepro32.dll	5.0.4518.0 160 KB	7/22/2002 12:05:04 PM			
PM	C:\WINNT\system32	Microsoft Corporation			
rsabase.dll	5.0.2195.3839	128 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
rsaenh.dll	5.0.2195.3839	131 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
rsapi32.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available
Available	Not Available				
rsasig.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available
Available	Not Available				
schannel.dll	5.1.2195.0 139 KB	7/22/2002 12:05:04 PM			
PM	C:\WINNT\system32	Microsoft Corporation			
shdoc401.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available
Available	Not Available				
shdocvw.dll	5.0.3502.5400	1079 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
shell32.dll	5.0.3502.5436	2319 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
shlwapi.dll	5.0.3502.5332	283 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
url.dll	5.0.3502.4510	82 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
urlmon.dll	5.0.3502.5400	442 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
vbscript.dll	5.6.0.6914 472 KB	9/14/2001 3:18:06 PM			
PM	C:\WINNT\system32	Microsoft Corporation			
webcheck.dll	5.0.3315.3727	251 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
win.com	5.0.2134.1 24 KB	12/7/1999 4:00:00 AM			
PM	C:\WINNT\system32	Microsoft Corporation			
wininet.dll	5.0.3502.4619	451 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
winsock.dll	3.10.0.103 3 KB	12/7/1999 4:00:00 AM			
PM	C:\WINNT\system32	Microsoft Corporation			
wintrust.dll	5.131.2195.3775	162 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
wsock.vxd	<File Missing>	Not Available	Not Available	Not Available	Not Available
Available	Not Available				
wsock32.dll	5.0.2195.4874	21 KB	7/22/2002 12:05:04		
PM	C:\WINNT\system32	Microsoft Corporation			
wsock32n.dll	<File Missing>	Not Available	Not Available	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	29279 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

RTE Input Parameters

3Tier.pro

Profile: 3T_24DE_12WC_24192wh
File Path: C:\Oly\Benchcraft\3T_24DE_12WC_24192wh.xml
Version: 4

Number of Engines: 48

Name: DRIVER01
Description:
Directory: c:\olylog.49.1
Machine: cl49
Parameter Set: Audit
Index: 0
Seed: 82204
Configured Users: 4680
Pipe Name: DRIVER1-1433709671
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER02
Description:
Directory: c:\olylog.49.2

Name: DRIVER03
Description:
Directory: c:\olylog.50.1
Machine: cl50
Parameter Set: Audit
Index: 20000000
Seed: 82204
Configured Users: 4680
Pipe Name: DRIVER3-1433643156
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER04
Description:
Directory: c:\olylog.50.2
Machine: cl50
Parameter Set: Audit
Index: 30000000
Seed: 82204
Configured Users: 4680
Pipe Name: DRIVER4-1433631734
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER05
Description:
Directory: c:\olylog.51.1
Machine: cl51
Parameter Set: Audit
Index: 40000000

Name: DRIVER06
Description:
Directory: c:\olylog.51.2
Machine: cl51
Parameter Set: Audit
Index: 50000000
Seed: 82204
Configured Users: 4680
Pipe Name: DRIVER6-1433579734
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER07
Description:
Directory: c:\olylog.52.1
Machine: cl52
Parameter Set: Audit
Index: 60000000
Seed: 82204
Configured Users: 4680
Pipe Name: DRIVER7-1433567468
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER08
Description:
Directory: c:\olylog.52.2
Machine: cl52
Parameter Set: Audit
Index: 70000000
Seed: 82204
Configured Users: 4680
Pipe Name: DRIVER8-1433555312

Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER09
Description:
Directory: c:\olylog.53.1
Machine: cl53
Parameter Set: Audit
Index: 80000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER9-1433543796
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER10
Description:
Directory: c:\olylog.53.2
Machine: cl53
Parameter Set: Audit
Index: 90000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER10-1433517406
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER11
Description:
Directory: c:\olylog.54.1
Machine: cl54
Parameter Set: Audit
Index: 100000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER11-1433487218
Connect Rate: 2000

Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER12
Description:
Directory: c:\olylog.54.2
Machine: cl54
Parameter Set: Audit
Index: 110000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER12-1433384468
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER13
Description:
Directory: c:\olylog.55.1
Machine: cl55
Parameter Set: Audit
Index: 120000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER13-1433372375
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER14
Description:
Directory: c:\olylog.55.2
Machine: cl55
Parameter Set: Audit
Index: 130000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER14-1433356953
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10

CLIENT_NURAND: 233
Additional Options:

Name: DRIVER15
Description:
Directory: c:\olylog.56.1
Machine: cl56
Parameter Set: Audit
Index: 140000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER15-1433338984
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER16
Description:
Directory: c:\olylog.56.2
Machine: cl56
Parameter Set: Audit
Index: 150000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER16-1433311234
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER17
Description:
Directory: c:\olylog.57.1
Machine: cl57
Parameter Set: Audit
Index: 160000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER17-1433291578
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER18
Description:
Directory: c:\olylog.57.2
Machine: cl57
Parameter Set: Audit
Index: 170000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER18-1433282015
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER19
Description:
Directory: c:\olylog.58.1
Machine: cl58
Parameter Set: Audit
Index: 180000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER19-1433271531
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER20
Description:
Directory: c:\olylog.58.2
Machine: cl58
Parameter Set: Audit
Index: 190000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER20-1433261875
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER21
Description:
Directory: c:\olylog.59.1
Machine: cl59
Parameter Set: Audit
Index: 200000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER21-1433248937
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER22
Description:
Directory: c:\olylog.59.2
Machine: cl59
Parameter Set: Audit
Index: 210000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER22-1433237843
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER23
Description:
Directory: c:\olylog.60.1
Machine: cl60
Parameter Set: Audit
Index: 220000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER23-1433209015
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER24
Description:
Directory: c:\olylog.60.2

Name: DRIVER25
Description:
Directory: c:\olylog.61.1
Machine: cl61
Parameter Set: Audit
Index: 230000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER24-1433053515
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER26
Description:
Directory: c:\olylog.61.2
Machine: cl61
Parameter Set: Audit
Index: 240000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER25-1433032390
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER27
Description:
Directory: c:\olylog.62.1
Machine: cl62
Parameter Set: Audit
Index: 250000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER26-1433019390
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER28
Description:
Directory: c:\olylog.62.2
Machine: cl62
Parameter Set: Audit
Index: 260000000

Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER27-1432990687
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER28
Description:
Directory: c:\olylog.62.2
Machine: cl62
Parameter Set: Audit
Index: 270000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER28-1432943468
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER29
Description:
Directory: c:\olylog.63.1
Machine: cl63
Parameter Set: Audit
Index: 280000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER29-1432916156
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER30
Description:
Directory: c:\olylog.63.2
Machine: cl63
Parameter Set: Audit
Index: 290000000
Seed: 82204

Name: DRIVER31
Description:
Directory: c:\olylog.64.1
Machine: cl64
Parameter Set: Audit
Index: 300000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER31-1432871171
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER32
Description:
Directory: c:\olylog.64.2
Machine: cl64
Parameter Set: Audit
Index: 310000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER32-1432851265
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER33
Description:
Directory: c:\olylog.41.1
Machine: cl41
Parameter Set: Audit
Index: 320000000
Seed: 82204
Configured Users: 4680
Pipe Name: DRIVER33-1335850906
Connect Rate: 2000

Name: DRIVER34
Description:
Directory: c:\olylog.41.2
Machine: cl41
Parameter Set: Audit
Index: 330000000
Seed: 82204
Configured Users: 4680
Pipe Name: DRIVER34-1335833484
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER35
Description:
Directory: c:\olylog.42.1
Machine: cl42
Parameter Set: Audit
Index: 340000000
Seed: 82204
Configured Users: 4680
Pipe Name: DRIVER35-1335773281
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER36
Description:
Directory: c:\olylog.42.2
Machine: cl42
Parameter Set: Audit
Index: 350000000
Seed: 82204
Configured Users: 4680
Pipe Name: DRIVER36-1335745468
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10

CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

Name: DRIVER37
 Description:
 Directory: c:\olylog.43.1
 Machine: cl43
 Parameter Set: Audit
 Index: 360000000
 Seed: 82204
 Configured Users: 5160
 Pipe Name: DRIVER37-1335720046
 Connect Rate: 2000
 Start Rate: 400
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

Name: DRIVER38
 Description:
 Directory: c:\olylog.43.2
 Machine: cl43
 Parameter Set: Audit
 Index: 370000000
 Seed: 82204
 Configured Users: 5160
 Pipe Name: DRIVER38-1335706718
 Connect Rate: 2000
 Start Rate: 400
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

Name: DRIVER39
 Description:
 Directory: c:\olylog.44.1
 Machine: cl44
 Parameter Set: Audit
 Index: 380000000
 Seed: 82204
 Configured Users: 5160
 Pipe Name: DRIVER39-1335689656
 Connect Rate: 2000
 Start Rate: 400
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND: 233

Additional Options:

Name: DRIVER40
 Description:
 Directory: c:\olylog.44.2
 Machine: cl44
 Parameter Set: Audit
 Index: 390000000
 Seed: 82204
 Configured Users: 5160
 Pipe Name: DRIVER40-1335672875
 Connect Rate: 2000
 Start Rate: 400
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

Name: DRIVER41
 Description:
 Directory: c:\olylog.45.1
 Machine: cl45
 Parameter Set: Audit
 Index: 400000000
 Seed: 82204
 Configured Users: 5160
 Pipe Name: DRIVER41-1335850906
 Connect Rate: 2000
 Start Rate: 400
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

Name: DRIVER42
 Description:
 Directory: c:\olylog.45.2
 Machine: cl45
 Parameter Set: Audit
 Index: 410000000
 Seed: 82204
 Configured Users: 5160
 Pipe Name: DRIVER42-1335833484
 Connect Rate: 2000
 Start Rate: 400
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

Name: DRIVER43
 Description:
 Directory: c:\olylog.46.1
 Machine: cl46
 Parameter Set: Audit
 Index: 420000000
 Seed: 82204
 Configured Users: 5160
 Pipe Name: DRIVER43-1335773281
 Connect Rate: 2000
 Start Rate: 400
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

Name: DRIVER44
 Description:
 Directory: c:\olylog.46.2
 Machine: cl46
 Parameter Set: Audit
 Index: 430000000
 Seed: 82204
 Configured Users: 5160
 Pipe Name: DRIVER44-1335745468
 Connect Rate: 2000
 Start Rate: 400
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND: 233
 CPU: 1
 Additional Options:

Name: DRIVER45
 Description:
 Directory: c:\olylog.47.1
 Machine: cl47
 Parameter Set: Audit
 Index: 440000000
 Seed: 82204
 Configured Users: 5160
 Pipe Name: DRIVER45-1335720046
 Connect Rate: 2000
 Start Rate: 400
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND: 233
 CPU: 0
 Additional Options:

Name: DRIVER46
 Description:
 Directory: c:\olylog.47.2

Machine: cl47
Parameter Set: Audit
Index: 450000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER46-1335706718
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: DRIVER47
Description:
Directory: c:\polylog.48.1
Machine: cl48
Parameter Set: Audit
Index: 460000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER47-1335689656
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER48
Description:
Directory: c:\polylog.48.2
Machine: cl48
Parameter Set: Audit
Index: 470000000
Seed: 82204
Configured Users: 5160
Pipe Name: DRIVER48-1335672875
Connect Rate: 2000
Start Rate: 400
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Number of User groups: 48

Driver Engine: DRIVER01
IIS Server: w25

Database: via:sqloly
User: sa
Protocol: HTML
w_id Range: 1 - 468
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 4680
District id: 1
Scale Down: No

Driver Engine: DRIVER02
IIS Server: w25
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 469 - 936
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 4680
District id: 1
Scale Down: No

Driver Engine: DRIVER03
IIS Server: w25
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 937 - 1404
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 4680
District id: 1
Scale Down: No

Driver Engine: DRIVER04
IIS Server: w25
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1405 - 1872
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 4680
District id: 1
Scale Down: No

Driver Engine: DRIVER05
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1873 - 2340
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 4680
District id: 1
Scale Down: No

Driver Engine: DRIVER06
IIS Server: w26
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2341 - 2808
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 4680
District id: 1
Scale Down: No

Driver Engine: DRIVER07
IIS Server: w26
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2809 - 3276
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 4680
District id: 1
Scale Down: No

Driver Engine: DRIVER08
IIS Server: w26
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3277 - 3744
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 4680
District id: 1

Scale Down: No

Driver Engine: DRIVER09
IIS Server: w27
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3745 - 4260
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER10
IIS Server: w27
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4261 - 4776
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER11
IIS Server: w27
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4777 - 5292
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER12
IIS Server: w27
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5293 - 5808
w_id Min Warehouse: 1

Scale: Normal
w_id Max Warehouse: 24192
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER13
IIS Server: w28
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5809 - 6324
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER14
IIS Server: w28
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 6325 - 6840
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER15
IIS Server: w28
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 6841 - 7356
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER16
IIS Server: w28
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML

w_id Min Warehouse: 7872
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER17
IIS Server: w29
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 7873 - 8388
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER18
IIS Server: w29
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 8389 - 8904
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER19
IIS Server: w29
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 8905 - 9420
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER20
IIS Server: w29
SQL Server: via:sqloly
Database: tpcc

User: sa
Protocol: HTML
w_id Range: 9421 - 9936
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER21
IIS Server: w30
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 9937 - 10452
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER22
IIS Server: w30
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 10453 - 10968
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER23
IIS Server: w30
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 10969 - 11484
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER24
IIS Server: w30
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 11485 - 12000
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER25
IIS Server: w31
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 12001 - 12516
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER26
IIS Server: w31
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 12517 - 13032
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER27
IIS Server: w31
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 13033 - 13548
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1

Scale Down: No
Driver Engine: DRIVER28
IIS Server: w31
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 13549 - 14064
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER29
IIS Server: w32
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 14065 - 14580
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER30
IIS Server: w32
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 14581 - 15096
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER31
IIS Server: w32
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 15097 - 15612
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal

User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER32
IIS Server: w32
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 15613 - 16128
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER33
IIS Server: w21
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 16129 - 16596
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 4680
District id: 1
Scale Down: No

Driver Engine: DRIVER34
IIS Server: w21
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 16597 - 17064
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 4680
District id: 1
Scale Down: No

Driver Engine: DRIVER35
IIS Server: w21
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML

w_id Range: 17065 - 17532
w_id Max Warehouse: 24192
Scale: Normal
User Count: 4680
District id: 1
Scale Down: No

Driver Engine: DRIVER36
IIS Server: w21
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 17533 - 18000
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 4680
District id: 1
Scale Down: No

Driver Engine: DRIVER37
IIS Server: w22
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 18001 - 18516
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER38
IIS Server: w22
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 18517 - 19032
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER39
IIS Server: w22
SQL Server: via:sqloly
Database: tpcc

Protocol: HTML
w_id Range: 19033 - 19548
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER40
IIS Server: w22
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 19549 - 20064
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER41
IIS Server: w23
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 20065 - 20580
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER42
IIS Server: w23
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 20581 - 21096
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER43
IIS Server: w23

SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 21097 - 21612
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER44
IIS Server: w23
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 21613 - 22128
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER45
IIS Server: w24
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 22129 - 22644
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER46
IIS Server: w24
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 22645 - 23160
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1

Scale Down: No
Driver Engine: DRIVER47
IIS Server: w24
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 23161 - 23676
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Driver Engine: DRIVER48
IIS Server: w24
SQL Server: via:sqloly
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 23677 - 24192
w_id Min Warehouse: 1
w_id Max Warehouse: 24192
Scale: Normal
User Count: 5160
District id: 1
Scale Down: No

Number of Parameter Sets: 1

		Audit					
		Audit parms					
Menu	Delay	Txn		Think	Key	RT	RT
		Weight	Time	Time	Delay	Fence	
	0.10	New Order	44.93	12.05		18.01	
		5.00	0.10				
		Payment	43.03	12.05		3.01	
		5.00	0.10				
		Delivery	4.01	5.05		2.01	
		5.00	0.10				
		Stock Level	4.02	5.05		2.01	
		20.00	0.10				
	0.10	Order Status	4.01	10.05		2.01	
		5.00	0.10				

Appendix D 60 Day Space Requirements

TPC-C 60 Day Space Requirements						
Warehouses	26,000				TpmC	301,225.52
Table	Rows	Data KB	Index KB	Extra 5% KB	8hr Space	Total Space KB
Warehouse	26,000	2,816	136	148		3100
District	260,000	29,352	2,184	1,577		33113
Customer	780,000,000	567,272,768	35,419,424	30,134,610		632826802
History	780,000,006	46,567,256	182,792		8,174,654	46750048
New_order	233,999,994	4,169,352	11,584	209,047		4389983
Orders	780,000,014	25,469,480	12,428,728		1,375,600	37898208
Order_line	2,147,483,647	520,001,056	1,245,960		96,425,468	521247016
Item	100,000	9,528	160	484		10172
Stock	2,147,483,647	832,000,008	1,782,792	41,689,140		875471940
Total		1,995,521,616	51,073,760	72,035,005	105,975,722	2,118,630,381
MB						
Dynamic Space	578,162	Sum of Data for Order, Orderline and History				
Static Space	1,490,813	Sum of Data+Index+5%-Dynamic Space				
Free Space	na	Total Allocated Spac - (Dynamic + Static Space)				
Daily Growth	107,174	(Dynamic Space/(W*62.5))*tpmc				
Daily Spread	-	(Free Space -1.5*Daily Growth) Zero Assumed				
60 Day Space MB	7,921,230					
60 Day Space GB	7,735.58	GB				
Log Size	300,000.00	MB				
KB Per New Order	4.69	KB				
8 hr log MB	661,886	MB				
8 hr log GB	646.3728	GB				
Space Usage	GB Needed	Disks Measured	GB Priced	Disk Size	Formatted Size	
60 Day Space DB	7,735.6	896	15,142.4	18.2GB	16.900	
			0.00			
			0.00			
Total DB			15142.40			
8-hr log + mirror	1,292.7	12	1,401.6	146	116.80	
OS, Swap	3	1	16.90	18.2GB	16.900	
Total Storage	9,031.32	GB	16,560.90	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

April 9, 2004

Hewlett-Packard
Company
Jason Goertz
14475 NE 24th St
Bellevue, WA 98007

Mr. Goertz:

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	16	\$264,656
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	12	\$8,856
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: PCjago0409042377

Please include this Reference ID in any correspondence regarding this price quote.

SANbox2 8 - switch by Qlogic - Fibre Adapters

[Qlogic - Fibre Adapters](#)



Our Price : \$4,861.58

FREE SHIPPING

Quantity:

1

[Add to cart](#)

Availability: 2

(As of 4/9/2004 12:50:26 PM)

[Email a friend](#)

[Add item to your list](#)

Make sure it's in stock?

[Check Real Time Availability](#)

PCU Part # : 219820 **Mfg. Part # :** SB2B-08A

Product Overview

SANbox2 8 - Switch - FC-AL, FC-PH, FC-PLDA, FC-FLA, FC-FG, FC-GS + 8 x SFP (empty) - 1 U

QLogic SANbox2 switches enable the highest-performance switched storage networking at affordable prices while protecting investments in 1Gb equipment. With SANbox2 switches, businesses can easily build powerful storage networks that are fully interoperable with any servers, networking equipment and storage subsystems. Certified to work with every major storage management application, SANbox2 switches are fabric-ready out of the box.

Based on the fifth-generation SANbox switch architecture, SANbox2 switches are the industry's fastest Fiber Channel switches that support 1Gb and 2Gb network performance. SANbox2 switches are ideal for new networks or entry-level, hub-based SANs seeking to substantially boost performance. With 8 or 16 ports for scalability, all SANbox2 switches are interoperable with any FC-SW-2 compliant switch.

This SANbox2-8 switch features front to back airflow.

Technical Specifications

Click Here for [Extended Specification](#)

Communication Mode	Full-duplex
Data Link Protocol	FC-AL, FC-PH, FC-PLDA, FC-FLA, FC-FG, FC-GS
Data Transfer Rate	2 Gbps
Device Type	Switch
Dimensions (WxDxH)	8.5 in x 19.7 in x 1.7 in
Empty Slots	8 x SFP (mini-GBIC)

Features	Uplink, manageable
Form Factor	External - 1 U
Power	AC 110/220 V (50/60 Hz)
Remote Protocol	Management SNMP, Telnet
Weight	11.5 lbs

 Hewlett Packard Company	HP Server rx8620		HP Sales Development 19111 Pruneridge Ave Cupertino, CA 95014 (408)-447-2320		
Description	Part Number	Unit Price	Qty	Extended Price	3 Yr Maint Price
rx8620 16-way SMP Base System					
incl cell boards, core IO, power supplies, PCI-X backplan, 16x1.5ghz Intel Itanium 2 6M processors	AB240A-002	\$273,995	1	\$273,995	
Support (Hardware & Software) 3yrs 24x7				\$0	\$117,292
8GB SDRAM (4x2GB DIMMs)	AB309A-OD1	\$14,000	16	\$224,000	
PCI-x I/O chassis	A6434A	\$29,995	1	\$29,995	
18gb 15K RPM disc for system boot/storage	A6540A	\$800	2	\$1,600	
Gig E - TX Adapter	A7061A	\$270	1	\$270	
Qlogic QLA-2350 Fibre-Channel VI Adapter	A390975	\$1,450	6	\$8,700	
LP9802 Fiber Channel Adapter	AB232A	\$1,558	2	\$3,116	
HP Smart Array Controller 5304	A9826A	\$2,399	16	\$38,384	
Storageworks enclosure 4414R	302969-B21	\$3,209	64	\$205,376	
Support 3yrs 24x7	171242-002	\$157	64		\$10,048
18GB, 15krpm Ultra3 Wide disk	286775-B22	\$299	896	\$267,904	
18GB, 15krpm Ultra3 Wide disk (10% spares)	286775-B22	\$299	90	\$26,910	
Modular Storage Array 1000	201723-B22	\$9,995	2	\$19,990	
Support 3yrs 24x7	402164-002	\$3,538	2		\$7,076
146GB, 10krpm disk	286716-B22	\$1,039	12	\$12,468	
146GB, 10krpm disk (10% spares)	286716-B22	\$1,039	2	\$2,078	
HP9000 Standard Rack System E41	A4902D	\$1,910	5	\$9,550	
5m LC to LC Cable Kit	221692-B22	\$82	2	\$164	
5m 2gbt Small Form Pluggable Adapter Kit	221470-B21	\$199	2	\$398	
HP Power Distribution Unit 120-240V	A5137AZ	\$145	6	\$870	
UPS R1500 XR	204404-001	\$866	1	\$866	
TA5300 Enclosure for DAT tape	C7508AZ	\$1,395	1	\$1,395	\$2,125
DVD Rom drive	C7499A	\$688	1	\$688	
DAT Tape	C7497A	\$1,532	1	\$1,532	
				\$1,130,249	\$136,541
Microsoft Windows Server 2003, Datacenter edition (64-bit)	T2372A opt016	\$32,000	1	\$32,000 incl. above	
				\$32,000	
HP ProLiant DL360 G3	308724-001	\$2,849.00	12	\$34,188	
2.80GHz Xeon Processor Option Kit (DL360 G3)	292892-B21	\$839.00	12	\$10,068	
HP Mouse	170299-B22	\$23.00	1	\$23	
HP Enhanced Keyboard	265977-001	\$12.00	1	\$12	
S5500 15 carbon / silver monitor	261602-001	\$129	1	\$129	
36.4-GB 15,000 rpm U320 Universal Hard Drive	286776-B22	\$519	12	\$6,228	
Qlogic QLA-2350 Fibre-Channel VI Adapter	A390975	\$1,450	12	\$17,400	
5M LC to LC Cable Kit	221692-B22	\$82	12	\$984	
5m 2gbt Small Form Pluggable Adapter Kit	221470-B21	\$199	12	\$2,388	
Support (Hardware & Software) 3yrs 24x7				\$71,420	\$34,104
				\$936,240	\$130,421
HP's Large Configuration Discount *	* Discounts:			-\$297,429	-\$40,224
	Total:			\$936,240	\$130,421