



i n v e n t

---

# TPC Benchmark® C Full Disclosure Report

---

HP Integrity rx6600

using Microsoft SQL Server 2005 Enterprise Itanium Edition  
SP1

on Microsoft Windows Server 2003, Enterprise Edition for  
Itanium-based Systems, SP1

First Edition  
July 18, 2006

First Edition - July 18, 2006

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 2006.

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., July 18, 2006

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 Integrity rx6600 in a client/server configuration, using Microsoft SQL Server 2005 Enterprise Itanium Edition SP1 and Microsoft COM+ Transaction Monitor. The operating system used for the benchmark was Microsoft Windows Server 2003, Enterprise Edition for Itanium-based Systems, SP1.

## TPC Benchmark® C Metrics

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


## Standard and Executive Summary Statements

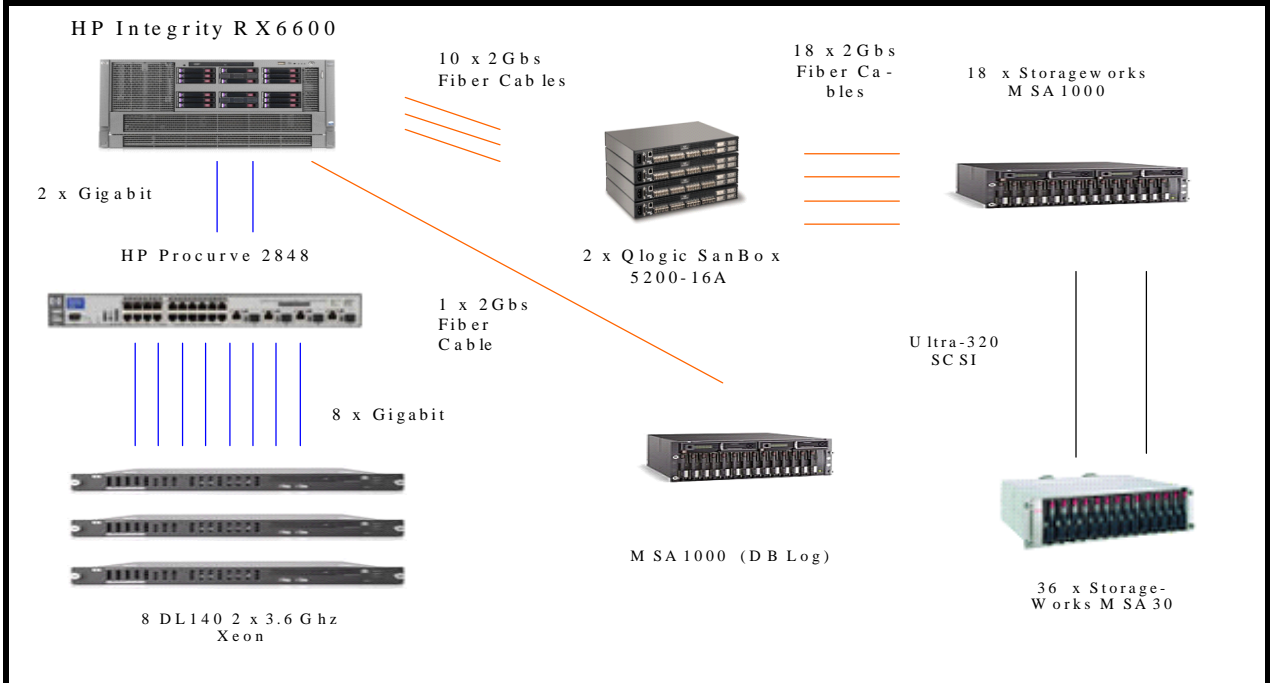
The following pages contain the executive summary of the benchmark results for the HP Integrity rx6600 system. The Standard System Summary is given below.

| Company Name            | System Name         | Database Software  | Operating System   |
|-------------------------|---------------------|--|--|
| Hewlett-Packard Company | HP Integrity rx6600 | Microsoft SQL Server 2005 Enterprise Itanium Edition SP1 | Microsoft Windows Server 2003, Enterprise Edition for Itanium-based Systems, SP1 |
| Total System Cost       | TPC-C Throughput    | Price/Performance  | Availability Date  |
| \$771,957 USD           | 344,928 tpmC        | \$2.24 USD per tpmC                                      | Dec 1, 2006  |


## 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.

|  |  |  |   |                          |
|--|--|--|---|--------------------------|
| <br><b>invent</b><br><b>Hewlett-Packard</b><br><b>Company</b> | <h1>HP Integrity rx6600</h1>                             |  |   | TPC-C Rev 5.7            |
|  |  |  |   | TPC-C Pricing Rev 1.1.0  |
| <b>Total System Cost</b>   | <b>TPC Throughput</b>                                    | <b>Price/Performance</b>   | <b>Report Date</b>  | <b>Availability Date</b> |
| <b>\$771,957 USD</b>   | <b>344,928 tpmC</b>                                      | <b>\$2.24 USD per tpmC</b>   | <b>July 18, 2006</b>  | <b>Dec 1, 2006</b>       |
| <b>Procs/Cores/Threads</b>   | <b>DataBase Manager</b>                                  | <b>Operating System</b>  | <b>Other Software</b>   | <b>Number of Users</b>   |
| Srvr - 4/8/16 Dual Core Itanium 2 Processor 9050 @ 1.6GHz Client - 8 x 2/2/4 Intel Xeon @ 3.6 GHz  | Microsoft SQL Server 2005 Enterprise Itanium Edition SP1 | Microsoft Windows Server 2003, Enterprise Edition for Itanium-based Systems, SP1 | Microsoft Visual C++<br>Microsoft COM+<br>Transaction Monitor | 276,000                  |



| System Components        | Server |   | Each Client |                    |
|--------------------------|--------|---|-------------|--------------------|
|                          | Qty    | Type                                    | Qty         | Type               |
| <b>Procs/Cores/Thrds</b> | 4/8/16 | Dual Core Itanium 2 Processor 9050 CPUs | 2/2/4       | 3.6 GHz Intel Xeon |
| <b>Cache Memory</b>      |        | 24 MB L3 cache                          |             | 512 KBYTE L2 Cache |
| <b>Memory</b>            | 12     | 4 x 4 Gbyte                             | 1           | 1024 MB            |
| <b>Disk Controllers</b>  | 5      | Qlogic 2342                             | 1           | ATA                |
|                          | 1      | Emulex 1050DC                           |             |                    |
| <b>Disk Drives</b>       | 756    | HP 36GB 15 KRPM                         | 1           | 80 Gbyte disk      |
|                          | 8      | HP 300GB 10 KRPM U320                   |             |                    |
|                          | 2      | HP 36 GB 10 KRPM SAS                    |             |                    |
| <b>Total Storage</b>     |        | 27625.00                                |             | 80 Gbyte           |
| <b>Tape Drives</b>       | 1      | HP TA5300                               |             |                    |
| <b>Terminals</b>         | 1      | Console Terminal                        | 1           | Console Terminal   |

|  <b>Hewlett Packard Company</b>   |           | <b>HP Integrity rx6600</b> |            | <b>TPCC Rev 5.7 April 2006 Pricing 1.1.0</b> |                      |                  |
|--|-----------|----------------------------|------------|--|----------------------|------------------|
| <b>July 18 2006</b>  |           |                            |            |  |                      |                  |
| Description  | Price Key | Part Numbr                 | Unit Price | Qty  | Extended Price       | 3 Yr Maint Price |
| HP Integrity rx6600 with (4) 1.6GHz/24MB Processor includes dual port 10/100/1000GbE adapter and 1 power supply  |           |                            |            |  |                      |                  |
| I/O backplane  | 1*        | AD134A#180                 | \$43,845   | 1  | \$43,845             |                  |
| core I/O: 8-port SAS Smart Array RAID Controller Card  | 1*        | AD296A                     | \$0        | 1  | \$0                  |                  |
| 192GB - 16GB DDR2 memory quad (4x4GB)  | 1*        | AB036A #100                | \$500      | 1  | \$500                |                  |
| 48 DIMM Carrier Board  | 1*        | AB566A                     | \$18,977   | 12   | \$227,724            |                  |
| 36GB, 10K rpm SAS HDD  | 1*        | AD127A                     | \$4,495    | 1  | \$4,495              |                  |
| Racked form factor kit   | 1*        | AD140A                     | \$382      | 2  | \$764                |                  |
| 3 Year Support (Hardware and Software)   | 1*        | AD053A                     | \$150      | 1  | \$150                |                  |
| DVD-ROM  | 1*        | HA110A3                    | \$10,947   | 1  |                      | \$10,947         |
| FC-HBA 2GB, 2 Channel (LP1050)   | 1         | AD142A                     | \$230      | 1  | \$230                |                  |
| Qlogic QLA-2342 Dual Port Fibre-Channel Adapter  | 1         | AD168A                     | \$2,000    | 1  | \$2,000              |                  |
| Qlogic QLA-2342 Dual Port Fibre-Channel Adapter (10% spares)   | 3         | QLA2342-CK                 | \$1,825    | 5  | \$9,125              |                  |
| 36GB, 15krpm Ultra320 disk   | 3         | QLA2342-CK                 | \$1,825    | 2  | \$3,650              |                  |
| 36GB, 15krpm Ultra320 disk (10% Spares)  | 1         | 286776-B22                 | \$269      | 756  | \$203,364            |                  |
| Storageworks MSA30 SB  | 1         | 286776-B22                 | \$269      | 76   | \$20,444             |                  |
| Storageworks MSA30 SB (10% Spares)   | 1         | 302969-B21                 | \$2,829    | 36   | \$101,844            |                  |
| HP Rack 5642   | 1         | 302969-B21                 | \$2,829    | 4  | \$11,316             |                  |
| UPS - HP R1500 XR Low Voltage US   | 1         | 358254-B21                 | \$689      | 5  | \$3,445              |                  |
| HP 16A High Voltage Modular PDU  | 1         | 204404-001                 | \$866      | 1  | \$866                |                  |
| Modular Storage Array 1000   | 1         | 252663-B24                 | \$299      | 10   | \$2,990              |                  |
| Modular Storage Array 1000 (10% spares)  | 1         | 201723-B22                 | \$6,995    | 19   | \$132,905            |                  |
| MSA1000 controller (10% spares)  | 1         | 201723-B22                 | \$6,995    | 2  | \$13,990             |                  |
| HP 300GB 10 KRPM U320  | 1         | 218231-B22                 | \$4,290    | 2  | \$8,580              |                  |
| HP 300GB 10 KRPM U320(10% Spares)  | 1         | 350964-B22                 | \$699      | 8  | \$5,592              |                  |
| 5M LC to LC Cable Kit  | 1         | 350964-B22                 | \$699      | 2  | \$1,398              |                  |
| TA5300 Enclosure for DAT tape  | 1         | 221692-B22                 | \$77       | 19   | \$1,463              |                  |
| DAT Tape   | 1         | C7508B                     | \$729      | 1  | \$729                |                  |
|  | 1         | C7497B                     | \$1,049    | 1  | \$1,049              |                  |
| <b>Server Subtotal</b>   |           |                            |            |  | <b>\$802,458</b>     | <b>\$10,947</b>  |
| Microsoft Windows Server 2003, Enterprise Edition for Itanium-based Systems, SP1   |           |                            |            |  |                      |                  |
|  | 1         | T2030A                     | \$2,149    | 1  | \$2,149              |                  |
| Microsoft SQL Server 2005 Enterprise Itanium Edition SP1   | 2         | 810-03134                  | \$23,432   | 4  | \$93,728             | \$245            |
| <b>Server Software Subtotal</b>  |           |                            |            |  | <b>\$95,877</b>      | <b>\$245</b>     |
| DL140 G2 3.6GHz Xeon 1GB 80GB SATA   |           |                            |            |  |                      |                  |
| 2nd 3.6GHz Xeon Processor for DL140  | 1         | 383504-001                 | \$2,149    | 8  | \$17,192             |                  |
| 3 Year Support (ProLiant Hardware)   | 1         | 378283-B21                 | \$999      | 8  | \$7,992              |                  |
| HP Mouse   | 1         | HA110A3                    | \$419      | 8  |                      | \$3,352          |
| HP Enhanced Keyboard (USB/PS2)   | 1         | P5304M                     | \$28       | 8  | \$224                |                  |
| HP ProCurve 2824 port switch   | 1         | DC852A#ABA                 | \$25       | 1  | \$25                 |                  |
| 3 Year Support (ProCurve Hardware)   | 1         | J4903A                     | \$2,499    | 1  | \$2,499              |                  |
| \$7540 17in CRT Monitor  | 1         | HA110A3                    | \$1,041    | 1  |                      | \$1,041          |
| Aten MasterView Plus 8 Port KVM Switch w/Cables  | 1         | PF997AA                    | \$139      | 1  | \$139                |                  |
|  | 4         | CS138A+                    | \$199      | 1  | \$199                |                  |
| <b>Client Subtotal</b>   |           |                            |            |  | <b>\$28,270</b>      | <b>\$4,393</b>   |
| Microsoft Windows 2003 Server  |           |                            |            |  |                      |                  |
| Microsoft Visual C++ Standard  | 2         | P73-00295                  | \$719      | 8  | \$5,752              |                  |
| Microsoft Problem Resolution   | 2         | 254-00170                  | \$109      | 1  | \$109                |                  |
|  | 2         |                            | \$245      | 1  |                      | \$245            |
| <b>Client Software Subtotal</b>  |           |                            |            |  | <b>\$5,861</b>       | <b>\$245</b>     |
| Qlogic SANBox 5200 16 2GB Port (w 16sfps, includes spares)   |           |                            |            |  |                      |                  |
| SANbox 5200 Prime 7x24x4 Maint. Upgrade, 1-Year  | 3         | SB5200-16A                 | \$5,192    | 2  | \$10,384             |                  |
|  | 5         | Prime-SB5200               | \$1,137    | 6  |                      | \$6,821          |
| <b>Connectivity Subtotal</b>   |           |                            |            |  | <b>\$10,384</b>      | <b>\$6,821</b>   |
| <b>**Total Extended Price:</b>   |           |                            |            |  | <b>\$965,501</b>     |                  |
| <b>**Total Discount:</b>   |           |                            |            |  | <b>-\$193,544</b>    |                  |
| <b>HP's Large Configuration Discount **</b>  |           |                            |            |  |                      |                  |
| Price Key: 1-HP, 2-Microsoft, 3-HBACentral.com 4 - KVMs.com 5 - Softwareforless.com  |           |                            |            |  |                      |                  |
| * Not immediately orderable. See Appendix E for details  |           |                            |            |  |                      |                  |
| ** A 35.41% discount was based on the overall value of the specific components from HP (Price Key 1) in this single quotation. Discounts for similarly sized configurations will be similar to those quoted here, but may vary based on the components in the configuration  |           |                            |            |  |                      |                  |
| <b>3 year cost of ownership:</b>   |           |                            |            |  | <b>\$771,957 USD</b> |                  |
| tpmC:  |           |                            |            |  | 344,928              |                  |
| <b>\$/tpmC:</b>  |           |                            |            |  | <b>\$2.24 USD</b>    |                  |
| 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 section of the TPC benchmark specification. If you find that the state prices are not available to these terms, please inform the TPC at pricing@tpc.org. Thank you |           |                            |            |  |                      |                  |
| Sales contact Vendor 1: HP Sales Development, 19111 Pruneridge Ave., Cupertino, CA 95014 (408) 447 2320 Sales contact<br>Vendor 2: Jamie Reding (425) 703-0510 jamiere@microsoft.com<br>Vendor 3 - HBACentral.com Vendor 4 - KVMs.com Vendor 5 - softwareforless.com   |           |                            |            |  |                      |                  |

## Numerical Quantities Summary for HP Integrity rx6600

**MQTH, Computed Maximum Qualified Throughput**

**344928 tpmC**

### Response Times (in seconds)

|                                | 90th %-ile | Maximum | Average |
|--------------------------------|------------|---------|---------|
| New-Order                      | 0.48s      | 5.81s   | 0.30s   |
| Payment                        | 0.46s      | 3.32s   | 0.28s   |
| Order-Status                   | 0.48s      | 5.37s   | 0.30s   |
| Delivery (interactive portion) | 0.13s      | 2.51s   | 0.11s   |
| Delivery (deferred portion)    | 0.13s      | 4.88s   | 0.09s   |
| Stock-Level                    | 0.69s      | 4.69s   | 0.46s   |
| Menu                           | 0.13s      | 3.02s   | 0.11s   |

Response time delay added for emulated components 0.1 seconds

### Transaction Mix, in percent of total transactions

|              |        |
|--------------|--------|
| New-Order    | 44.95% |
| Payment      | 43.02% |
| Delivery     | 4.01%  |
| Stock-Level  | 4.02%  |
| Order-Status | 4.00%  |

### Keying/Think Times

|                        | Keying Time |        |        | Think Time |        |         |
|------------------------|-------------|--------|--------|------------|--------|---------|
|                        | Min         | Avg    | Max    | Min        | Avg    | Max     |
| New-Order              | 18.00s      | 18.02s | 18.08s | 0.00s      | 12.06s | 120.55s |
| Payment                | 3.00s       | 3.02s  | 3.08s  | 0.00s      | 12.07s | 120.53s |
| Order-Status           | 3.00s       | 3.02s  | 3.08s  | 0.00s      | 10.06s | 100.53s |
| Delivery (interactive) | 3.00s       | 3.02s  | 3.08s  | 0.00s      | 5.07s  | 50.53s  |
| Stock-Level            | 3.00s       | 3.02s  | 3.08s  | 0.00s      | 5.06s  | 50.53s  |

### Test Duration

|  |             |
|--|-------------|
| Ramp up time                             | 22 minutes  |
| Measurement interval                     | 120 minutes |
| Transactions during measurement interval | 95773416    |
| Ramp down time                           | 5 minutes   |

### Checkpointing

|   |               |
|---|---------------|
| Number of checkpoints in measurement interval | 4             |
| Checkpoint Interval                           | 28.33 minutes |

# Table of Contents

|  |          |
|--|----------|
| Abstract.....  | 3        |
| Overview.....  | 3        |
| TPC Benchmark® C Metrics .....                         | 3        |
| Standard and Executive Summary Statements .....        | 3        |
| Auditor.....   | 3        |
| <b>Table of Contents.....</b>                          | <b>7</b> |
| Preface.....   | 9        |
| Document Structure .....                               | 9        |
| TPC Benchmark® C Overview .....                        | 9        |
| System Overview .....                                  | 10       |
| General Items.....                                     | 11       |
| Test Sponsor .....                                     | 11       |
| Application Code and Definition Statements.....        | 11       |
| Parameter Settings .....                               | 11       |
| Configuration Diagrams .....                           | 11       |
| Chapter 1 Logical Database Design .....                | 13       |
| 1.1 Table Definitions .....                            | 13       |
| 1.2 Physical Organization of the Database.....         | 13       |
| 1.3 Insert and Delete Operations.....                  | 13       |
| 1.4 Partitioning.....                                  | 13       |
| 1.5 Replication, Duplication or Additions.....         | 13       |
| Chapter 2 Transaction and Terminal Profiles.....       | 14       |
| 2.1 Random Number Generation .....                     | 14       |
| 2.2 Input/Output Screen Layout.....                    | 14       |
| 2.3 Priced Terminal Feature Verification.....          | 14       |
| 2.4 Transaction Statistics.....                        | 14       |
| 2.5 Presentation Manager or Intelligent Terminal ..... | 15       |
| 2.6 Queuing Mechanism .....                            | 15       |
| Chapter 3 Transaction and System Properties .....      | 16       |
| 3.1 Transaction System Properties (ACID Tests).....    | 16       |
| 3.2 Atomicity Tests .....                              | 16       |
| 3.2.1 COMMIT Transaction.....                          | 16       |
| 3.2.2 ROLLBACK Transaction .....                       | 16       |
| 3.3 Consistency Tests .....                            | 16       |
| 3.4 Isolation Tests .....                              | 17       |
| 3.5 Durability Tests.....                              | 17       |
| 3.5.1 Loss of Data / Loss of Log .....                 | 17       |
| 3.5.2 Loss of System / Memory .....                    | 18       |
| Chapter 4 Scaling and Database Population.....         | 19       |
| 4.1 Database Layout .....                              | 19       |
| 4.2 Initial Cardinality of Tables .....                | 24       |
| 4.3 60 Day Space .....                                 | 24       |
| 4.3.1 Transaction Log Space Requirements.....          | 24       |
| 4.4 Type of Database Used.....                         | 25       |
| 4.5 Database Mapping.....                              | 25       |
| Chapter 5 Performance Metrics and Response Time.....   | 26       |
| 5.1 Throughput .....                                   | 26       |
| 5.2 Response Times .....                               | 26       |
| 5.3 Keying and Think Times .....                       | 26       |
| 5.4 Response Time Frequency .....                      | 27       |
| 5.4.1 New Order Response Time.....                     | 27       |
| 5.4.2 Payment Response Time Distribution.....          | 28       |

|            |  |     |
|------------|--|-----|
| 5.4.3      | Order Status Response Time .....                             | 29  |
| 5.4.4      | Delivery Response Time Distribution.....                     | 30  |
| 5.4.5      | Stock Level Response Time .....                              | 31  |
| 5.4.6      | Response Time Versus Throughput.....                         | 32  |
| 5.4.7      | New Order Think Time Distribution .....                      | 33  |
| 5.4.8      | Throughput Versus Time Distribution .....                    | 34  |
| 5.5        | Steady State Determination.....                              | 34  |
| 5.6        | Work Performed During Steady State.....                      | 34  |
| 5.6.1      | Checkpoint.....  | 35  |
| 5.6.2      | Checkpoint Conditions .....                                  | 35  |
| 5.6.3      | Checkpoint Implementation.....                               | 35  |
| 5.7        | Measurement Period Duration .....                            | 35  |
| 5.8        | Regulation of Transaction Mix .....                          | 35  |
| 5.9        | Transaction Mix.....   | 35  |
| 5.10       | Transaction Statistics.....                                  | 36  |
| 5.11       | Checkpoint Count and Location.....                           | 36  |
| Chapter 6  | SUT, Driver and Communications Definition .....              | 37  |
| 6.1        | RTE Description .....  | 37  |
| 6.2        | Emulated Components .....                                    | 37  |
| 6.3        | Functional Diagram .....                                     | 37  |
| 6.4        | Networks .....   | 37  |
| 6.5        | Operator Intervention .....                                  | 37  |
| Chapter 7  | Pricing.....   | 38  |
| 7.1        | System Pricing.....  | 38  |
| 7.2        | General Availability, Throughput and Price Performance ..... | 38  |
| 7.3        | Country Specific Pricing.....                                | 38  |
| 7.4        | Usage Pricing.....   | 38  |
| Chapter 8  | Audit .....  | 40  |
| 8.1        | Auditor's Information .....                                  | 40  |
| Appendix A | Source Code.....   | 43  |
| Appendix B | Database Load .....  | 174 |
| B.1        | Database Options.....  | 202 |
| B.2        | Table definitions .....                                      | 203 |
| B.3        | Stored Procedures.....                                       | 207 |
| Appendix C | Tunable Parameters .....                                     | 213 |
| C.1        | Microsoft SQL Server 8.0 Configuration Parameters.....       | 250 |
| C.2        | Client System Configuration Parameters .....                 | 251 |
| C.3        | RTE Input Parameters.....                                    | 279 |
| Appendix D | 60 Day Space Requirements.....                               | 285 |
| Appendix E | 3 <sup>rd</sup> Party Pricing .....                          | 286 |



# Preface

## Document Structure

This is the full disclosure report for a benchmark test of the HP Integrity rx6600 using Microsoft SQL Server 2005 Enterprise Itanium Edition SP1. It meets the requirements of the TPC Benchmark® C Standard Specification, Revision 5.7 dated April 2006. TPC Benchmark® C was developed by the Transaction Processing Performance Council (TPC). It is the intent of this group to develop a suite of benchmarks to measure the performance of computer systems executing a wide range of applications. Hewlett-Packard Company and Microsoft, Inc. are active participants in the TPC.

## TPC Benchmark® C Overview

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

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

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

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

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

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

## System Overview

The hardware configuration used in this TPC-C test was based on the HP Integrity rx6600. 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, Enterprise Edition for Itanium-based Systems, SP1 and the database was Microsoft SQL Server 2005 Enterprise Itanium Edition SP1, Version 9.00.2047.00.

The processor architecture of the HP Integrity rx6600 was designed for the Dual Core Itanium 2 Processor 9050 processor. The HP Integrity rx6600 used in this test was powered by 4 1.6GHz Dual Core Itanium 2 Processor 9050 processors, each with 24 MB of 3rd level cache. The 4 processors contain 8 cores, and threading was enabled. This configuration therefore presented 16 logical processors to the operating system.

This configuration used 192 GB of HP SDRAM.

The operating system, all executables and libraries, the master database, and swap space were contained in one 36GB hard disk, attached to the internal SCSI controller. A partition was created on the same disc array as the log and was used for utility storage of scripts, the build environment, etc.

The database log drive storage was located on 1 HP MSA1000. The MSA1000 held 8 HP 300GB 10 KRPM U320 hard drives. The MSA1000 disk array was connected to the HP Integrity rx6600 using 1 Emulex 1050 Fibrechannel HBA. The disks were configured as RAID 1+0, and two battery backed up disk array controller caches were enabled and mirrored on each MSA1000 (90% write, 10% read).

The TPC-C database storage consisted of 756 HP 36GB 15 KRPM disk drives. 18 HP StorageWorks MSA1000 disk arrays were used to connect the disks. Each MSA1000 holds 14 discs. Connected to each MSA1000 via 2 Ultra-320 SCSI channels were two StorageWorks MSA30's. Each MSA30 holds also holds 14 disks. The discs were configured as a RAID0 Array spanning all 42 discs in each array, for a total of 18 RAID0 database arrays. The MSA1000 caches were disabled on the RAID0 partitions. Additionally, a RAID1+0 volume was configured on each set of 42 discs as fault tolerant backup for the database. Windows partitions were created on the RAID0 volumes to contain the CS and MISC SQL filegroups. The partition sizes were the same on all 18 volumes.

Each of the 8 clients is an HP Proliant DL140 G2 with 2 Intel Xeon Processors at 3.6 GHz, 1024 MB RAM and one 80 GB ATA hard disk, running Microsoft Windows 2003 Server with IIS 6. Threading was enabled, so 4 logical processors were presented to the operating system.

The server and web-clients were networked together using standard Gigabit LAN connections. 12 remote terminal emulators (RTEs) emulated 276,000 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 Gigabit mode. HP DL140's were also used as the RTE emulators.

Microsoft SQL Server 2005 Enterprise Itanium Edition SP1 was configured to utilize "soft NUMA", a feature that allows network connections to be affined to specific groups of CPUs (The HP Integrity rx6600 has no hardware NUMA capability). SQL Server was configured with 8 SoftNuma nodes, each configured with 2 CPUs each. The Checkpoint process was affined to the 8<sup>th</sup> Soft NUMA node, and that node's workload was reduced to allow for the CPU needed for the checkpoint. A script doing continuous checkpoints of 1750 seconds (29 minutes, 10 seconds) was started on one of the web servers after steady state was reached with a connection port that connected to the 8th Numa Node. This allowed the main checkpoint process to run on that single processor, which directed the processing of the checkpoint tasks that SQL assigned to each of the other 8 soft Numa Nodes.

## 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.

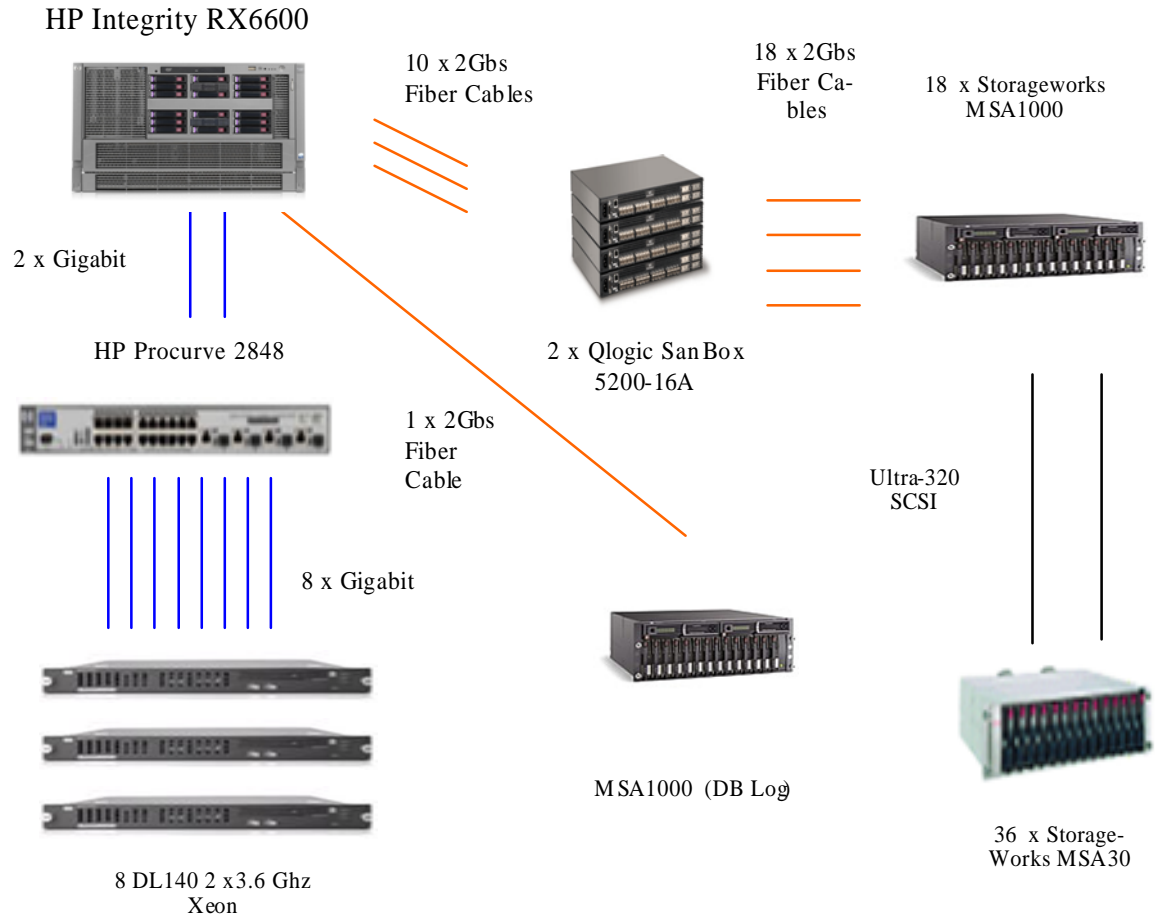
Appendix D contains the 60 day space calculations.

### 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**



# 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 766 disk drives, 756 HP 36GB 15 KRPM , 8 HP 300GB 10 KRPM U320 drives for log, and 2 HP 36 GB 10 KRPM SAS(s) for the operating system.

Part of the space on each of the 18 database disk arrays was configured as 1 RAID0 volume over 42 36GB 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 18 Arrays was configured as a RAID1+0 volume over all 42 36GB drives. Each volume contained backups of the database.

## 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           | 85.00% |
|                 | Remote warehouse         | 15.00% |
|                 | Non primary key access   | 59.99% |
| Order Status    | Non primary key access   | 60.06% |
| Delivery        | Skipped transactions     | 0      |
| Transaction Mix | New Order                | 44.95% |
|                 | Payment                  | 43.02% |
|                 | Delivery                 | 4.01%  |
|                 | Stock Level              | 4.02%  |
|                 | Order Status             | 4.00%  |

## 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 Integrity rx6600. 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 ROLLBACK 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 tests 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 27,600 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 27,600 (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. Transactions 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 30,000 warehouses in a single test. The standard driving mechanism was used to generate the transaction load of 276,000 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. 276,000 users were logged in to the database and ran transactions.
4. Rampup was performed until the TPMC rate was 90% of the reported rate.
5. After 5 minutes, one of the (mirrored) log disk was removed from the system, processing transactions continued.
6. After another 5 minutes, the system was reset using the built-in Maintenance Processor. This reset the hardware, reran memory initialization, and reloaded the Windows OS.
7. The RTE continued running and completed transactions enroute from the clients were recorded. Error messages began appearing on the RTE screen.
8. The RTE was stopped.
9. After Windows was finished booting, 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 5 Qlogic 2342 Disk Array Controllers. The MSA1000 disk array that held the database log was connected to an Emulex 1050 FC HBA.

The measured database configuration used a total of 766 disks, which included 756 HP 36GB 15 KRPM drives for data, 8 HP 300GB 10 KRPM U320 drives for log, and 2 HP 36 GB 10 KRPM SAS drive(s) for the operating system. Part of the space on each of the 18 database disk arrays was configured as 1 RAID0 volume over 42 36GBdrives. 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 18 Arrays was configured as a RAID1+0 volume over all 42 36 GB drives. Each volume contained backups of the database.

Table 2a shows the complete data distribution.

The database log drive storage was located on 1 HP MSA1000. The MSA1000 held 8 HP 300GB 10 KRPM U320 hard drives. The MSA1000 disk array were connected to the HP Integrity rx6600 using 1 Emulex 1050 Fibrechannel HBA. The disks were configured as RAID 1+0, and two battery backed up disk array controller caches were enabled and mirrored on each MSA1000 (90% write, 10% read). Not all the space visible to the OS on the log array was configured for the log in SQL, but could have been if needed. Table 2b shows the log distribution.

#### **Table 2a: Data Distribution**

|                               |          |      |   |                             |
|-------------------------------|----------|------|---|-----------------------------|
| Modular Storage Array 1000 #1 |          |      | WINDOWS.NET DISK ADMIN                            |                             |
|                               |          |      | DISK 1  |                             |
| SCSI ID                       | Channels |      | Partitions (RAID 0): 157GB Disk                   |                             |
|                               | 0        | 1    | 0   | 1                           |
| 0                             | 36GB     | 36GB | G:\MNT\CS1<br>Raw<br>101GB                        | G:\MNT\MISC1<br>Raw<br>56GB |
| 1                             | 36GB     | 36GB |   |                             |
| 2                             | 36GB     | 36GB |   |                             |
| 3                             | 36GB     | 36GB |   |                             |
| 4                             | 36GB     | 36GB |   |                             |
| 5                             | 36GB     | 36GB |   |                             |
| 8                             | 36GB     | 36GB |   |                             |
| MSA1000 U320 Channels         |          |      | WINDOWS.NET DISK ADMIN                            |                             |
| 2 StorageWorks MSA30          |          |      | DISK 2  |                             |
| SCSI ID                       | Channels |      | Partitions (RAID 1+0): 633GB                      |                             |
|                               | 2        | 3    | 0   |                             |
| 0                             | 36GB     | 36GB | G:\MNT\BACKUP\1\<br>NTFS Volume Backup1<br>320 GB |                             |
| 1                             | 36GB     | 36GB |   |                             |
| 2                             | 36GB     | 36GB |   |                             |
| 3                             | 36GB     | 36GB |   |                             |
| 4                             | 36GB     | 36GB |   |                             |
| 5                             | 36GB     | 36GB |   |                             |
| 8                             | 36GB     | 36GB |   |                             |
| 9                             | 36GB     | 36GB |   |                             |
| 10                            | 36GB     | 36GB |   |                             |
| 11                            | 36GB     | 36GB |   |                             |
| 12                            | 36GB     | 36GB |   |                             |
| 13                            | 36GB     | 36GB |   |                             |
| 14                            | 36GB     | 36GB |   |                             |
| 15                            | 36GB     | 36GB |   |                             |

|                               |          |      |  |                             |
|-------------------------------|----------|------|--|-----------------------------|
| Modular Storage Array 1000 #2 |          |      | WINDOWS.NET DISK ADMIN                           |                             |
|                               |          |      | DISK 3   |                             |
| SCSI ID                       | Channels |      | Partitions (RAID 0): 157GB Disk                  |                             |
|                               | 0        | 1    | 0  | 1                           |
| 0                             | 36GB     | 36GB | G:\MNT\CS2<br>Raw<br>101GB                       | G:\MNT\MISC2<br>Raw<br>57GB |
| 1                             | 36GB     | 36GB |  |                             |
| 2                             | 36GB     | 36GB |  |                             |
| 3                             | 36GB     | 36GB |  |                             |
| 4                             | 36GB     | 36GB |  |                             |
| 5                             | 36GB     | 36GB |  |                             |
| 8                             | 36GB     | 36GB |  |                             |
| MSA1000 U320 Channels         |          |      | WINDOWS.NET DISK ADMIN                           |                             |
| 2 StorageWorks MSA30          |          |      | DISK 4   |                             |
| SCSI ID                       | Channels |      | Partitions (RAID 1+0): 633GB                     |                             |
|                               | 2        | 3    | 0  |                             |
| 0                             | 36GB     | 36GB | G:\MNT\BACKUP\2<br>NTFS Volume Backup2<br>130 GB |                             |
| 1                             | 36GB     | 36GB |  |                             |
| 2                             | 36GB     | 36GB |  |                             |
| 3                             | 36GB     | 36GB |  |                             |
| 4                             | 36GB     | 36GB |  |                             |
| 5                             | 36GB     | 36GB |  |                             |
| 8                             | 36GB     | 36GB |  |                             |
| 9                             | 36GB     | 36GB |  |                             |
| 10                            | 36GB     | 36GB |  |                             |
| 11                            | 36GB     | 36GB |  |                             |
| 12                            | 36GB     | 36GB |  |                             |
| 13                            | 36GB     | 36GB |  |                             |
| 14                            | 36GB     | 36GB |  |                             |
| 15                            | 36GB     | 36GB |  |                             |

.....

....

| Modular Storage Array 1000 #18                  |          |      | WINDOWS.NET DISK ADMIN                             |                              |
|---|----------|------|--|------------------------------|
|   |          |      | DISK 35  |                              |
| SCSI ID   | Channels |      | Partitions (RAID 0): 157GB Disk                    |                              |
|   | 0        | 1    | 0  | 1                            |
| 0   | 36GB     | 36GB | G:\MNT\CS18<br>Raw<br>101GB                        | G:\MNT\MISC18<br>Raw<br>57GB |
| 1   | 36GB     | 36GB |  |                              |
| 2   | 36GB     | 36GB |  |                              |
| 3   | 36GB     | 36GB |  |                              |
| 4   | 36GB     | 36GB |  |                              |
| 5   | 36GB     | 36GB |  |                              |
| 8   | 36GB     | 36GB |  |                              |
| MSA1000 U320 Channels<br>2 StorageWorks MSA30's |          |      |  |                              |
|   |          |      | DISK 36  |                              |
| SCSI ID   | Channels |      | Partitions (RAID 1+0): 633GB                       |                              |
|   | 2        | 3    | 0  |                              |
| 0   | 36GB     | 36GB | G:\MNT\BACKUP\18<br>NTFS Volume Backup24<br>130 GB |                              |
| 1   | 36GB     | 36GB |  |                              |
| 2   | 36GB     | 36GB |  |                              |
| 3   | 36GB     | 36GB |  |                              |
| 4   | 36GB     | 36GB |  |                              |
| 5   | 36GB     | 36GB |  |                              |
| 8   | 36GB     | 36GB |  |                              |
| 9   | 36GB     | 36GB |  |                              |
| 10  | 36GB     | 36GB |  |                              |
| 11  | 36GB     | 36GB |  |                              |
| 12  | 36GB     | 36GB |  |                              |
| 13  | 36GB     | 36GB |  |                              |
| 14  | 36GB     | 36GB |  |                              |
| 15  | 36GB     | 36GB |  |                              |

**Table 2b: Log Distribution**

| Emulex 1050 Fiber Adapter #1 |          |      | WINDOWS.NET DISK ADMIN                                 |                         |                     |
|------------------------------|----------|------|--|-------------------------|---------------------|
| 1 MSA1000                    |          |      | DISK 1 (Dynamic)                                       |                         |                     |
| SCSI ID                      | Channels |      | Partitions (RAID 1+0 1117.46 GB)                       |                         |                     |
|                              | 0        | 1    | 0  | 1                       | 2                   |
| 0                            | 36GB     | 36GB | (no drv ltr)<br>Raw<br>10 MB<br>(for proper alignment) | L:<br>Raw<br>1092.46 GB | G:<br>NTFS<br>25 GB |
| 1                            | 36GB     | 36GB |  |                         |                     |
| 2                            | 36GB     | 36GB |  |                         |                     |
| 3                            | 36GB     | 36GB |  |                         |                     |
| 4                            | 36GB     | 36GB |  |                         |                     |
| 5                            | 36GB     | 36GB |  |                         |                     |
| 8                            | 36GB     | 36GB |  |                         |                     |

## 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  | 30,000        |
| District   | 300,000       |
| Customer   | 900,000,000   |
| History    | 900,000,000   |
| Orders     | 900,000,000   |
| New Orders | 270,000,000   |
| Order Line | 840,041,944   |
| Stock      | 3,000,000,000 |
| 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 60day 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/I, 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 2005 Enterprise Itanium Edition SP1 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. cs\_fg consists of 18 partitions at 103300 MB each and misc\_fg consist of 18 partitions at 57300 MB each as shown in the createdb.sql. The log was configured with 400,000 MB at database creation, and was expanded to 1,500,000 MB after database creation and load.

## Chapter 5 Performance Metrics and Response Time

### 5.1 Throughput

Measured tpmC® must be reported.

Measured TpmC®: 344,928  
Price per TpmC®: \$2.24 USD

### 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                      | 0.30s   | 0.48s      | 5.81s   |
| Payment                        | 0.28s   | 0.46s      | 3.32s   |
| Order-Status                   | 0.30s   | 0.48s      | 5.37s   |
| Delivery (interactive portion) | 0.11s   | 0.13s      | 2.51s   |
| Delivery (deferred portion)    | 0.09s   | 0.13s      | 4.88s   |
| Stock-Level                    | 0.46s   | 0.69s      | 4.69s   |
| Menu                           | 0.11s   | 0.13s      | 3.02s   |

### 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            | 18.00   | 18.02s  | 18.08s  |
| Payment              | 3.00    | 3.02s   | 3.08s   |
| Order Status         | 3.00    | 3.02s   | 3.08s   |
| Interactive Delivery | 3.00    | 3.02s   | 3.08s   |
| Stock Level          | 3.00    | 3.02s   | 3.08s   |

**Table 6: Transaction Think Times**

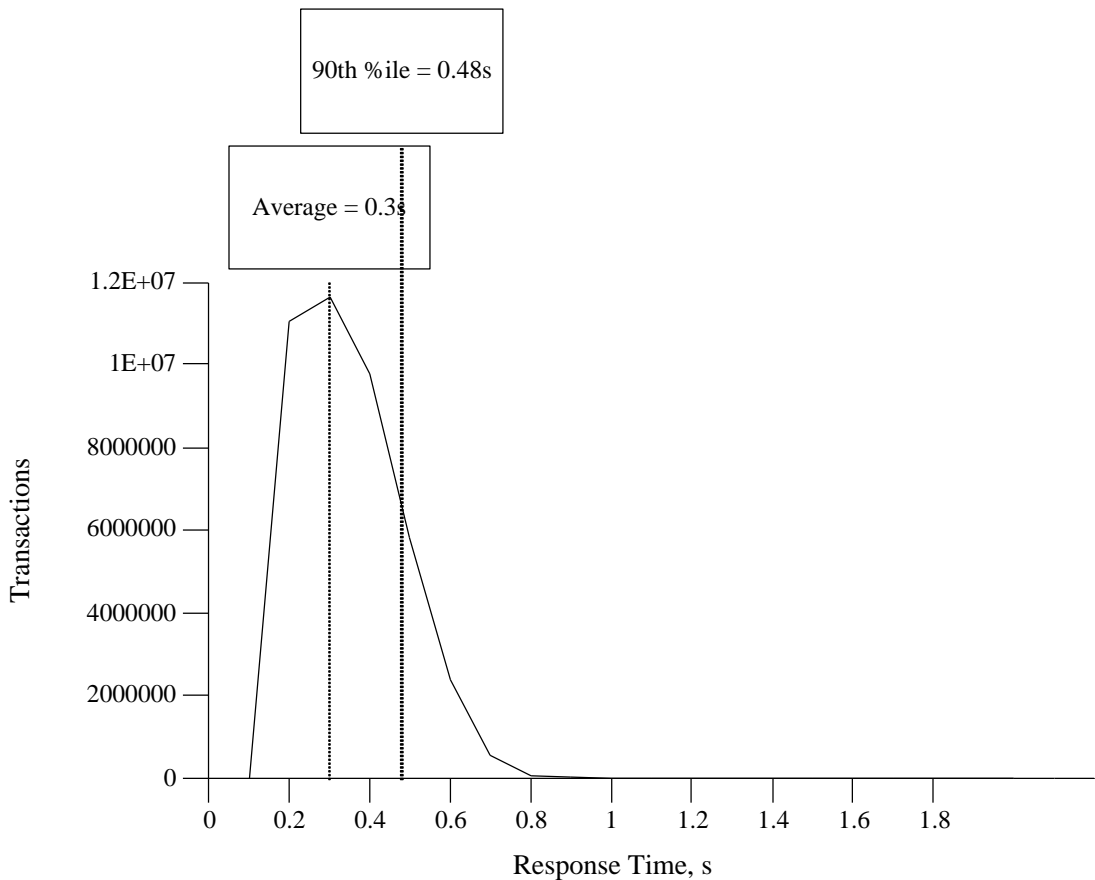
| Think Times          | Minimum | Average | Maximum |
|----------------------|---------|---------|---------|
| New Order            | 0       | 12.06s  | 120.55s |
| Payment              | 0       | 12.07s  | 120.53s |
| Order Status         | 0       | 10.06s  | 100.53s |
| Interactive Delivery | 0       | 5.07s   | 50.53s  |
| Stock Level          | 0       | 5.06s   | 50.53s  |

## 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

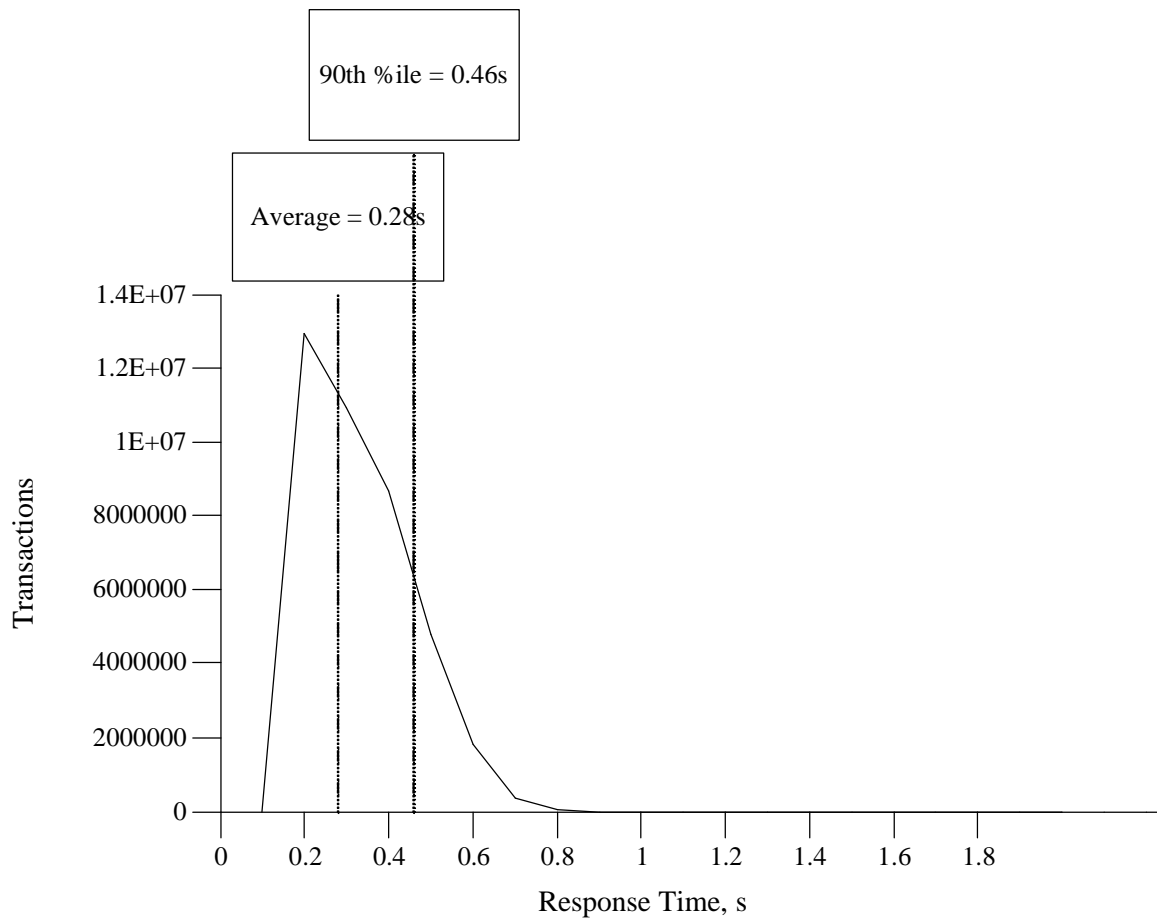
Figure 3: New Order Response Time Distribution



Response time frequency distribution for New Order transaction

## 5.4.2 Payment Response Time Distribution

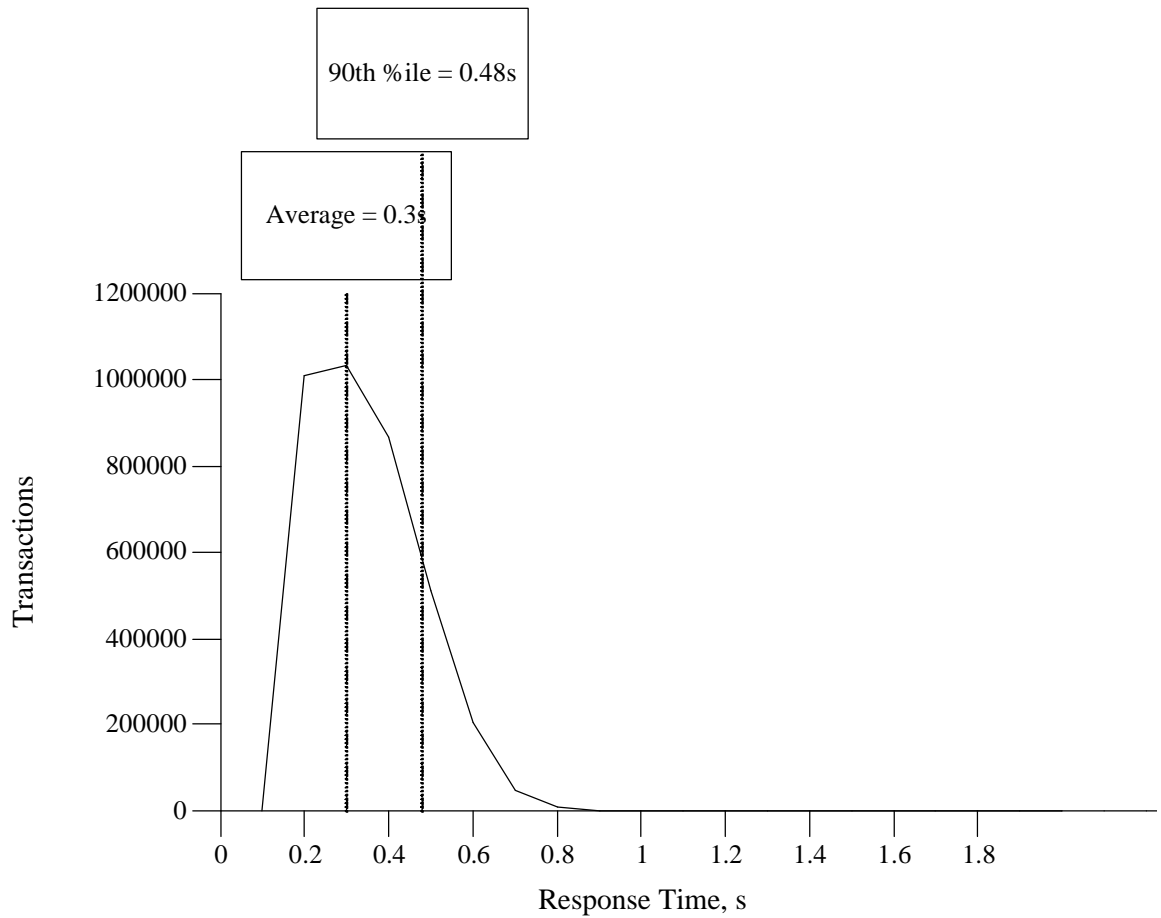
Figure 4: Payment Response Time Distribution



Response time frequency distribution for Payment transaction

### 5.4.3 Order Status Response Time

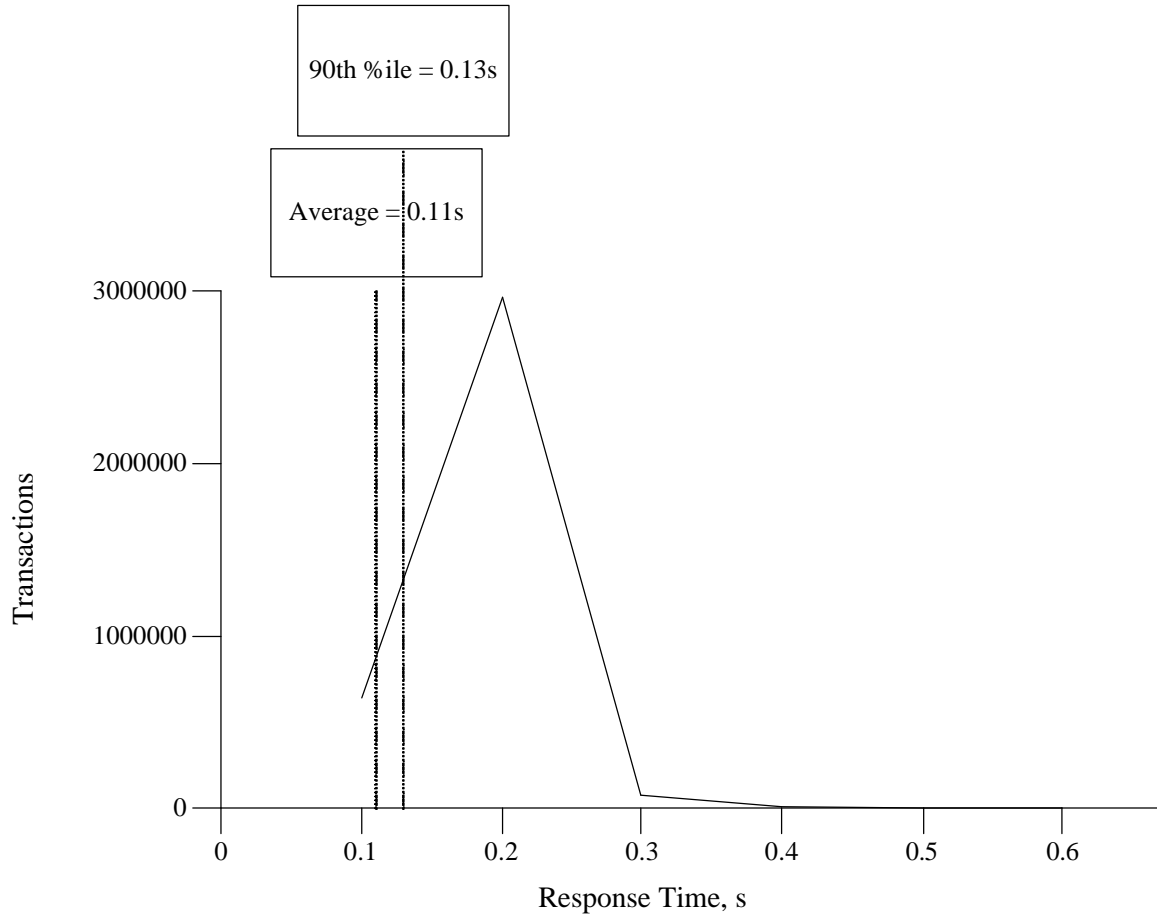
Figure 5: Order Status Response Time Distribution



Response time frequency distribution for Order Status transaction

## 5.4.4 Delivery Response Time Distribution

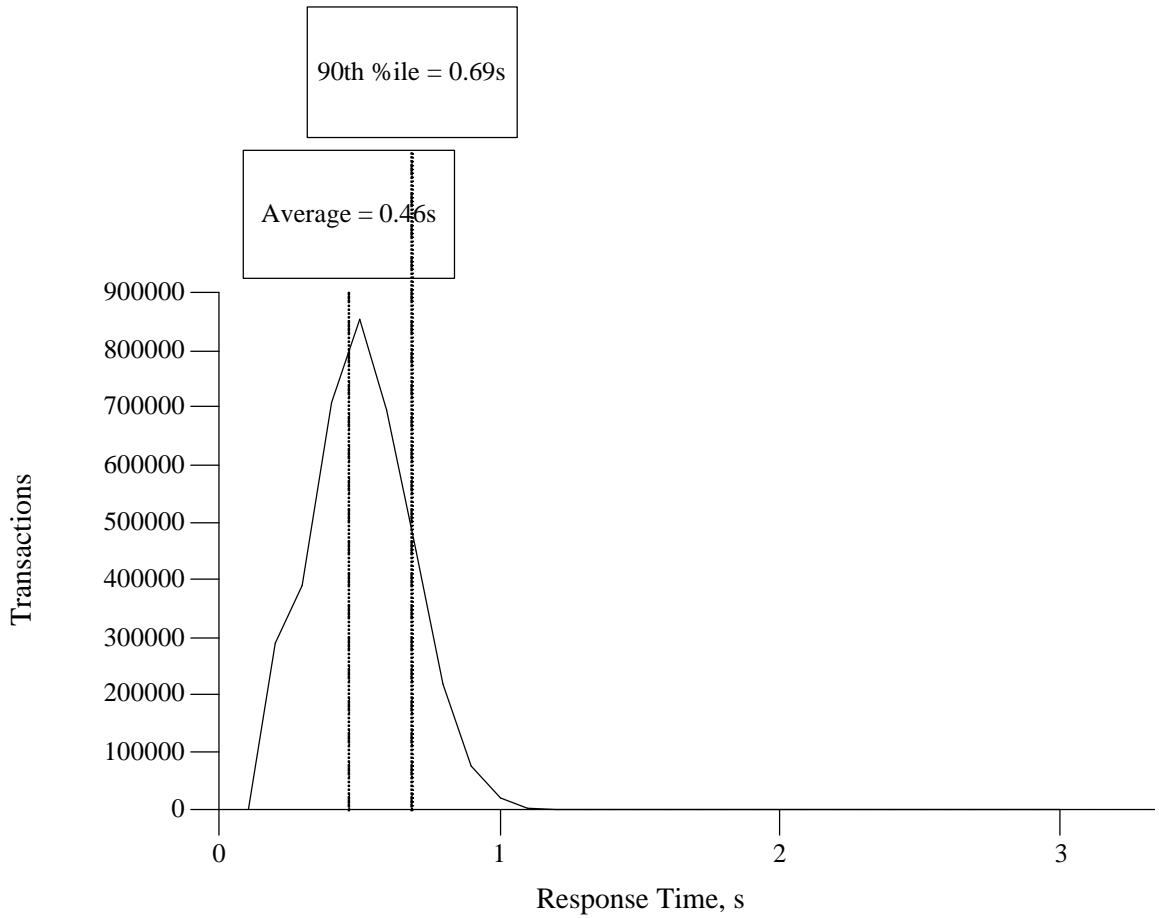
Figure 6: Delivery Response Time Distribution



Response time frequency distribution for Delivery transaction

5.4.5 Stock Level Response Time

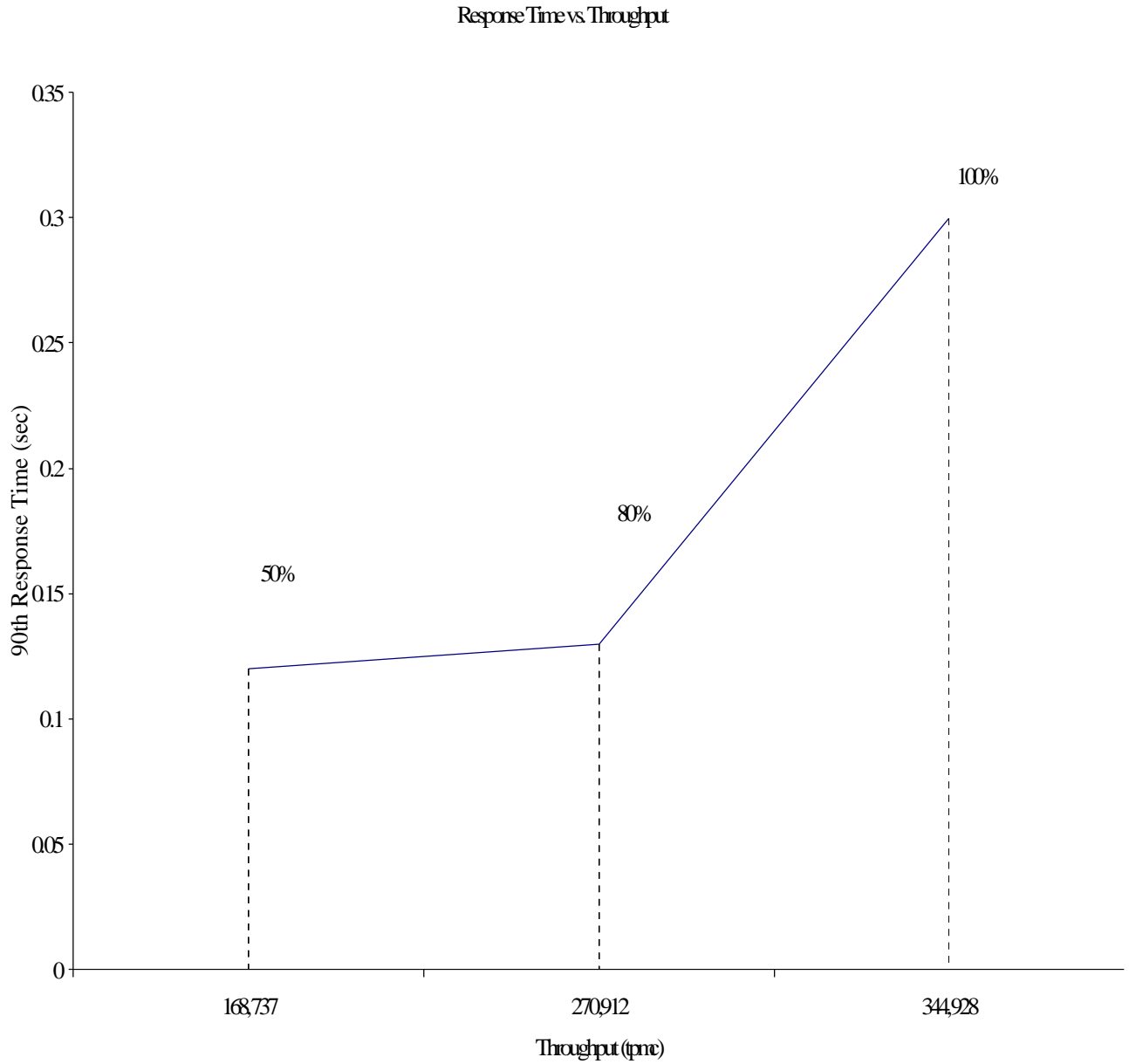
Figure 7: Stock Level Response Time Distribution



Response time frequency distribution for Stock Level transaction

## 5.4.6 Response Time Versus Throughput

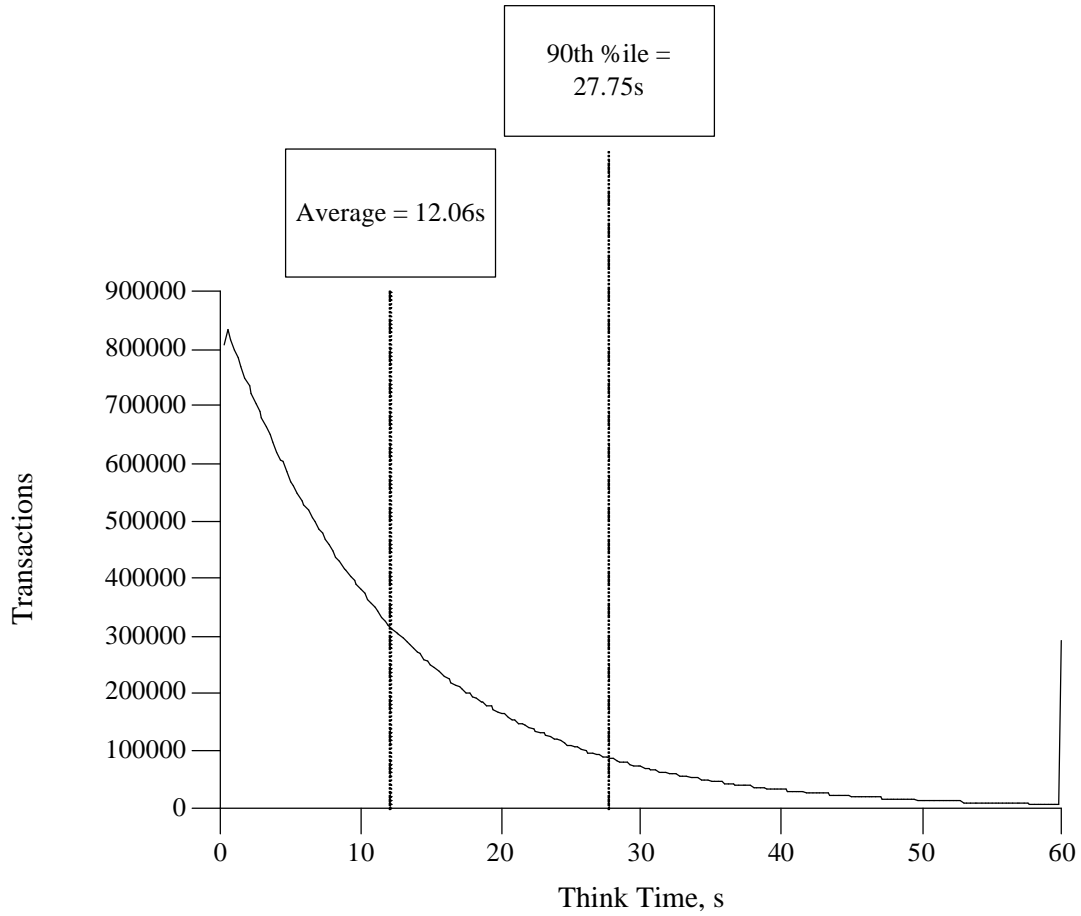
Figure 8: New Order Response Time Distribution





5.4.7 New Order Think Time Distribution

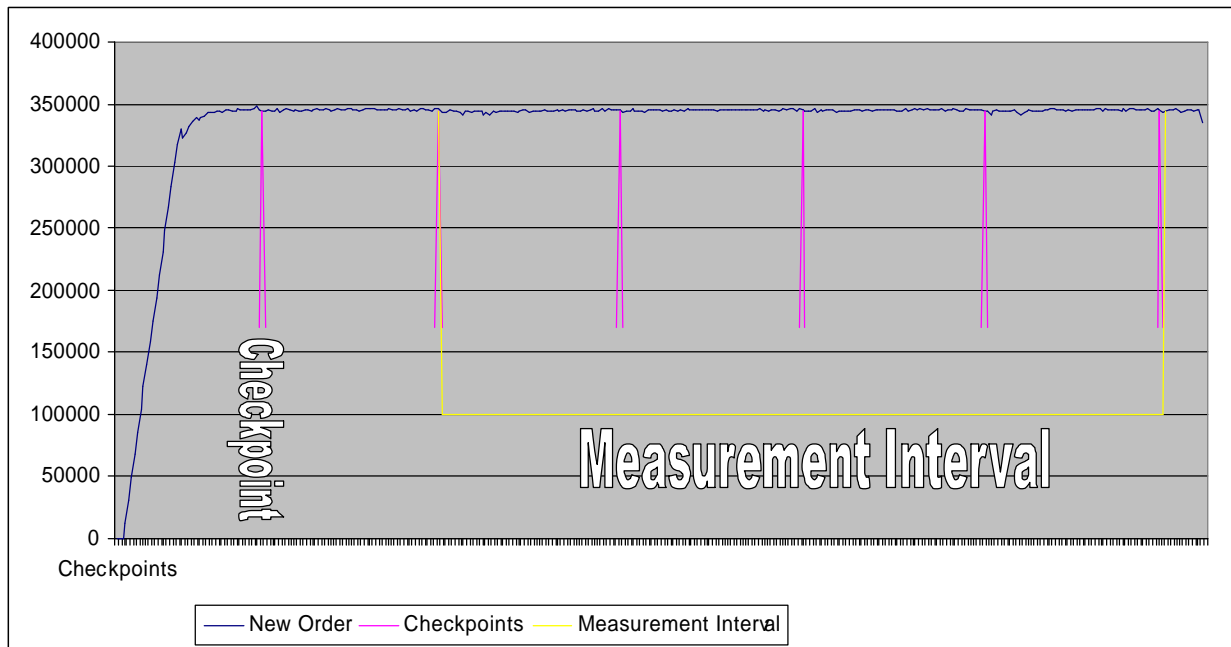
Figure 9: New Order Think Time Distribution



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. The "CHECKPOINT 1750" syntax in Microsoft SQL Server 2005 Enterprise Itanium Edition SP1 was used to force the checkpoints to an interval of 29 minutes, 10 seconds. The checkpoints were affinityized to a 2 processor SoftNuma node using tcp connection affinity. The script was run on the one of the web clients. 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 2005 Enterprise Itanium Edition SP1 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 2005 Enterprise Itanium Edition SP1 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    | 44.95%     |
| Payment      | 43.02%     |
| Delivery     | 4.01%      |
| Stock Level  | 4.02%      |
| Order Status | 4.00%      |

## 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 4 checkpoints within the measurement interval and 2 checkpoint before the measurement interval.

**Table 8: Measurement Interval and Checkpoints**

| Event             | From     | To       |
|-------------------|----------|----------|
| Measured Interval | 12:22:52 | 14:22:52 |
| Checkpoint        | 11:53:00 | 12:22:10 |
| Checkpoint        | 12:22:52 | 12:52:02 |
| Checkpoint        | 12:52:44 | 13:21:54 |
| Checkpoint        | 13:22:35 | 13:51:45 |
| Checkpoint        | 13:52:26 | 14:21:36 |

## 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.*

A Gigabit network was used between the RTEs and the clients, another Gigabit network 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**

|                                      |                     |
|--------------------------------------|---------------------|
| <b>Maximum qualified throughput:</b> | 344,928 tpmC        |
| <b>Price per tpmC:</b>               | \$2.24 USD per tpmC |
| <b>Availability:</b>                 | Dec 1, 2006         |

### 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:

- 4 Microsoft SQL Server 2005 Enterprise Itanium Edition SP1 per-processor licenses.
- Microsoft Windows Server 2003, Datacenter edition (64-bit)
- 8 Microsoft Windows 2003 Server licenses.
- 1 Microsoft Visual C++ 32bit Edition.

- 3 year support for hardware components

## **7.5 Testing**

*This is a test*

## 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  
PO Box 984  
140 Klamath Blvd  
Klamath, CA 95548  
(707) 482-0523  
Fax (707) 482-0575

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

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

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





**PERFORMANCE METRICS INC.**  
**TPC Certified Auditors**

July 17, 2006

Mr. Eric Deehr  
Performance Engineer  
Windows Integrity Engineering  
Hewlett-Packard Company  
14475 NE 24<sup>th</sup> St.  
Bellevue, WA 98007

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

Platform: HP 6600  
Database Manager: Microsoft SQL Server 2005 Enterprise Itanium Edition  
Operating System: Microsoft Windows Server 2003, Enterprise Edition 64 bit  
Transaction Monitor: Microsoft COM+

| Server:                             |            |                           |              |                |
|-------------------------------------|------------|---------------------------|--------------|----------------|
| CPU's                               | Memory     | Disks (total)             | 90% Response | TpmC           |
| 4 Intel @<br>1.6 Ghz                | 192 GB     | 8 @ 300 GB<br>756 @ 36 GB | 0.48         | <b>344,928</b> |
| 8 Clients: ProLiant DL25 each with: |            |                           |              |                |
| 2 Intel @ 3.6<br>Ghz                | Main: 1 GB | 1 @ 80 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 30,000 warehouses, of which 27,600 were active during the measured interval.
- Inactive warehouses were verified to be unchanged during the performance run.
- The ACID properties were successfully demonstrated.
- 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 60day space calculation was verified.
- The Measured cycle times were confirmed to have the correct response time delays.
- There were 276,000 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.
- 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:  
None

Sincerely,



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
 *
 * Microsoft TPC-C Kit Ver. 4.20.000
 *
 * Copyright Microsoft, 1999
 * All Rights Reserved
 *
 * Version 4.10.000 audited by
 * Richard Gimarc, Performance Metrics, 3/17/99
 *
 * PURPOSE: Header file for ISAPI
 * TPCC.DLL, defines structures and functions used
 * in the isapi tpcc.dll.
 */
//VERSION RESOURCE DEFINES
#define _APS_NEXT_RESOURCE_VALUE 101
#define _APS_NEXT_COMMAND_VALUE 40001
#define _APS_NEXT_CONTROL_VALUE 1000
#define _APS_NEXT_SYMED_VALUE 101
#define TP_MAX_RETRIES 50
//note that the welcome form must be processed
//first as terminal ids assigned here, once the
//terminal id is assigned then the forms can be
//processed in any order.
#define WELCOME_FORM 1
//beginning form no term id
//assigned, form id
#define MAIN_MENU_FORM 2
//term id assigned main menu form
id
```

```
#define NEW_ORDER_FORM 3
//new order form id
#define PAYMENT_FORM 4
//payment form id
#define DELIVERY_FORM 5
//delivery form id
#define ORDER_STATUS_FORM 6 //order
status id
#define STOCK_LEVEL_FORM 7 //stock
level form id
//This macro is used to prevent the compiler
//error unused formal parameter
#define UNUSEDPARAM(x) (x = x)
//This structure defines the data necessary to
//keep distinct for each terminal or client
//connection.
typedef struct _CLIENTDATA
{
    int iNextFree; //index of next free
    element or -1 if this entry in use.
    int w_id; //warehouse id assigned at
    welcome form
    int d_id; //district
    id assigned at welcome form
    int iSyncId; //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_RAN
    GE,
    ERR_DELIVERY_CARRIER_INVALID,
    ERR_DELIVERY_MISSING_OCD_KEY
    Y,
    ERR_DELIVERY_THREAD_FAILED,
    ERR_GETPROCADDR_FAILED,
    ERR_HTML_ILL_FORMED,
    ERR_INVALID_SYNC_CONNECTION
    ,
    ERR_INVALID_TERMID,
    ERR_LOADDLL_FAILED,
    ERR_MAX_CONNECTIONS_EXCEED
    ED,
    ERR_MEM_ALLOC_FAILED,
    ERR_MISSING_REGISTRY_ENTRIES
    ,
    ERR_NEWORDER_CUSTOMER_INV
    ALID,
    ERR_NEWORDER_CUSTOMER_KEY,
    ERR_NEWORDER_DISTRICT_INVAL
    ID,
    ERR_NEWORDER_FORM_MISSING_
    DID,
    ERR_NEWORDER_ITEMID_INVALID
    ,
    ERR_NEWORDER_ITEMID_RANGE,
    ERR_NEWORDER_ITEMID_WITHOU
    T_SUPPW,
    ERR_NEWORDER_MISSING_IID_KEY
    Y,
    ERR_NEWORDER_MISSING_QTY_K
    EY,
    ERR_NEWORDER_MISSING_SUPPW
    _KEY,
```

```

ED,
ERR_NEWORDER_NOITEMS_ENTER
ERR_NEWORDER_QTY_INVALID,
ERR_NEWORDER_QTY_RANGE,
SUPPW,
ERR_NEWORDER_QTY_WITHOUT_
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_INVALI
D,
ERR_PAYMENT_CWI_INVALID,
ERR_PAYMENT_DISTRICT_INVALID
ERR_PAYMENT_HAM_INVALID,
ERR_PAYMENT_HAM_RANGE,
ONG,
ERR_PAYMENT_LAST_NAME_TO_L
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_KE
Y,

```

```

ERR_STOCKLEVEL_MISSING_THRE
SHOLD_KEY,
ERR_STOCKLEVEL_THRESHOLD_IN
VALID,
ERR_STOCKLEVEL_THRESHOLD_RA
NGE,
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_WEBDDL;};
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_BLO
CK *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 *szForm);
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_B
LOCK *pECB, int iTermId, char
*szBuffer);
void
ProcessPaymentForm(EXTENSION_CONTROL_BL
OCK *pECB, int iTermId, char *szBuffer);

```

```

void
ProcessOrderStatusForm(EXTENSION_CONTROL_
BLOCK *pECB, int iTermId, char *szBuffer);

void
ProcessDeliveryForm(EXTENSION_CONTROL_B
OCK *pECB, int iTermId, char *szBuffer);

void
ProcessStockLevelForm(EXTENSION_CONTROL_B
LOCK *pECB, int iTermId, char *szBuffer);
void GetNewOrderData(LPSTR lpszQueryString,
NEW_ORDER_DATA *pNewOrderData);
void GetPaymentData(LPSTR lpszQueryString,
PAYMENT_DATA *pPaymentData);
void GetOrderStatusData(LPSTR lpszQueryString,
ORDER_STATUS_DATA *pOrderStatusData);
BOOL PostDeliveryInfo(long w_id, short
o_carrier_id);
BOOL IsNumeric(char *ptr);
BOOL IsDecimal(char *ptr);
void DeliveryWorkerThread(void *ptr);

```

### Isapi\_dll/src/tpcc.rc

```

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

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

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

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

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

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

VS_VERSION_INFO VERSIONINFO
FILEVERSION 0,4,0,0
PRODUCTVERSION 0,4,0,0
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x40004L

```

```

FILESVERSIONINFO
BEGIN
BLOCK "StringFileInfo"
BEGIN
BLOCK "040904b0"
BEGIN
VALUE "Comments", "TPC-C HTML DLL
Server (DBLIB)0"
VALUE "CompanyName", "Microsoft0"
VALUE "FileDescription", "TPC-C HTML
DLL Server (DBLIB)0"
VALUE "FileVersion", "0, 4, 0, 00"
VALUE "InternalName", "tpcc0"
VALUE "LegalCopyright", "Copyright ©
19970"
VALUE "OriginalFilename", "tpcc.dll0"
VALUE "ProductName", "Microsoft
tpcc0"
VALUE "ProductVersion", "0, 4, 0, 00"
END
END
BLOCK "VarFileInfo"
BEGIN
VALUE "Translation", 0x409, 1200
END
END

#ifdef !_MAC

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

1 TEXTINCLUDE DISCARDABLE
BEGIN
"resource.h0"
END

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

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

#endif // APSTUDIO_INVOKED

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

IDD_DIALOG1 DIALOG DISCARDABLE 0, 0, 186,
95
STYLE DS_MODALFRAME | WS_POPUP |
WS_CAPTION | WS_SYSTEMU
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
////////////////////////////////////
////////////////////////////////////
#endif

#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 "420"

```

```

static CRITICAL_SECTION
TermCriticalSection;
static HINSTANCE hLibInstanceTm = NULL;
static HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB
*pCTPCC_DBLIB_new;

TYPE_CTPCC_ODBC
*pCTPCC_ODBC_new;

TYPE_CTPCC_TUXEDO
*pCTPCC_TUXEDO_new;
TYPE_CTPCC_ENCINA *pCTPCC_ENCINA_new;
TYPE_CTPCC_ENCINA
*pCTPCC_ENCINA_post_init;
TYPE_CTPCC_COM
*pCTPCC_COM_new;

// For deferred Delivery txns:

CTxnLog
*txnDelilog = NULL;
//used

to log delivery transaction information

HANDLE
hWorkerSemaphore
= INVALID_HANDLE_VALUE;

HANDLE
hDoneEvent
=
INVALID_HANDLE_VALUE;
HANDLE
*pDelihandles
= NULL;

// configuration settings from registry
TPCCREGISTRYDATA Reg;

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

DWORD
dwDelBuffSize
= 100;
// size of circular buffer for delivery
txns
DWORD
dwDelBuffFreeCount;
// number of buffers free

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

// Critical section to synchronize connection open
and close.
//
CRITICAL_SECTION hConnectCriticalSection;

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

```

```

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

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

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

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

GetComputerName(szMyComputerName,
&dwSize);

szMyComputerName[dwSize] = 0;
}

DisableThreadLibraryCalls((HMODULE)hModule);

InitializeCriticalSection(&TermCriticalSection);

if ( ReadTPCCRegistrySettings(
&Reg ) )

```

```

        throw new
CWEBCLNT_ERR(
ERR_MISSING_REGISTRY_ENTRIES);

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

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

        TermInit();

        // load DLL for txn monitor
        if (Reg.eTxnMon == TUXEDO)
        {
                strcpy( szDllName,
Reg.szPath );

                strcat( szDllName,
"tpcc_tuxedo.dll");

                hLibInstanceTm =
LoadLibrary( szDllName );

                if (hLibInstanceTm
== NULL)

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

                // get function pointer
to wrapper for class constructor

                pCTPCC_TUXEDO_new =
(TYPE_CTPCC_TUXEDO*)
GetProcAddress(hLibInstanceTm,"CTPCC_TUXED
O_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_ENCIN
A_new");

                pCTPCC_ENCINA_post_init =
(TYPE_CTPCC_ENCINA*)
GetProcAddress(hLibInstanceTm,"CTPCC_ENCIN
A_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_n
ew");

                                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)
{
// Initialize delivery
delay critical section

//

InitializeCriticalSection(&hConnectC
riticalSection);

// for deferred
delivery txns:

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

InitializeCriticalSection(&DelBuffCritic
alSection);

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
);

wprintf( szLogFile,
"%sdelivery-%2.2d%2.2d%2.2d-%2.2d%2.2d-%2.2d-%2.2d-%2.2dms.log",

```

```

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

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(&DelBuffCritic
alSection);

// Delete delivery
delay critical section

//

DeleteCriticalSection(&hConnectCritic
alSection);

}

DeleteCriticalSection(&TermCriticalS
ection);

if (hLibInstanceTm != NULL)

FreeLibrary(
hLibInstanceTm );

hLibInstanceTm = NULL;

if (hLibInstanceDb != NULL)

FreeLibrary(
hLibInstanceDb );

hLibInstanceDb = NULL;

Sleep(500);

```



```

break;

default:

/* nothing */;
}
}
catch (CBaseErr *e)
{
    TCHAR szMsg[256];

    _sntprintf(szMsg,
sizeof(szMsg), "%s error, code %d: %s",
e->ErrorTypeStr(), e->ErrorNum(),
e->ErrorText());

    WriteMessageToEventLog( szMsg );
    delete e;

    TerminateExtension(0);
    return FALSE;
}
catch (...)
{

    WriteMessageToEventLog(TEXT("U
nhandled 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
K *pECB)
{
    int
iCmd, FormId, TermId, iSyncId;
    char
szBuffer[4096];

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

```

```

char
szHeader1[4096];

#ifdef ICECAP
StartCAP();
#endif

try
{
    //process http query

    ProcessQueryString(pECB, &iCmd,
&FormId, &TermId, &iSyncId);

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

            // debugging...

            char szTmp[128];

            wsprintf( szTmp, "Invalid term ID;
TermId = %d", TermId );

            WriteMessageToEventLog( szTmp );

            throw new CWEBCLNT_ERR(
ERR_INVALID_TERMID );
        }

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

        throw new CWEBCLNT_ERR(
ERR_INVALID_SYNC_CONNECTION );

        //set use
time

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

    switch(iCmd)
    {
    case 0:

        WelcomeForm(pECB, szBuffer);
        break;

    case 1:

        switch(
FormId )
        {

        case WELCOME_FORM:

        case MAIN_MENU_FORM:

            break;

        case NEW_ORDER_FORM:

            ProcessNewOrderForm(pECB,
TermId, szBuffer);

```

```

        break;
    case PAYMENT_FORM:
        ProcessPaymentForm(pECB,
            TermId, szBuffer);
        break;
    case DELIVERY_FORM:
        ProcessDeliveryForm(pECB, TermId,
            szBuffer);
        break;
    case ORDER_STATUS_FORM:
        ProcessOrderStatusForm(pECB,
            TermId, szBuffer);
        break;
    case STOCK_LEVEL_FORM:
        ProcessStockLevelForm(pECB,
            TermId, szBuffer);
        break;
    }
    break;
case 2:
    // new-
    order selected from menu; display new-order
    input form
    MakeNewOrderForm(TermId, NULL,
        INPUT_FORM, szBuffer);
    break;
case 3:
    //
    payment selected from menu; display payment
    input form
    MakePaymentForm(TermId, NULL,
        INPUT_FORM, szBuffer);
    break;
case 4:
    //
    delivery selected from menu; display delivery
    input form
    MakeDeliveryForm(TermId, NULL,
        INPUT_FORM, szBuffer);
    break;
case 5:
    // order-
    status selected from menu; display order-status
    input form
    MakeOrderStatusForm(TermId,
        NULL, INPUT_FORM, szBuffer);
    break;
case 6:
    // stock-
    level selected from menu; display stock-level
    input form

```

```

        MakeStockLevelForm(TermId,
        NULL, INPUT_FORM, szBuffer); break;
        case 7:
            //
            ExitCmd
            TermDelete(TermId);
            WelcomeForm(pECB, szBuffer);
            break;
        case 8:
            SubmitCmd(pECB, szBuffer);
            break;
        case 9:
            // menu
            MakeMainMenuForm(TermId,
            Term.pClientData[TermId].iSyncId, szBuffer);
            break;
        case 10:
            //
            CMD=Clear
            // resets
            all connections; should only be used when no
            other connections are active
            TermDeleteAll();
            TermInit();
            WelcomeForm(pECB, szBuffer);
            break;
        case 11:
            //
            CMD=Stats
            StatsCmd(pECB, szBuffer);
            break;
    }
    catch (CBaseErr *e)
    {
        ErrorForm( pECB, e-
        >ErrorType(), e->ErrorNum(), TermId, iSyncId,
        e->ErrorText(), szBuffer);
        delete e;
    }
    catch (...)
    {
        ErrorForm( pECB,
        ERR_TYPE_WEBDLL, 0, TermId, iSyncId, "Error:
        Unhandled exception in Web Client.", szBuffer );
    }
#ifdef ICECAP
    StopCAP();
#endif
    lpbSize = strlen(szBuffer);
    wprintf(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
        (LPTSTR *)lpszStrings, // array of
        error strings
        NULL); // no raw data
    }

    (VOID)
    DeregisterEventSource(hEventSource);
}

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

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

    DELIVERY_TRANSACTION
    delivery;

```

```

PDELIVERY_DATA
pDeliveryData;
TXN_RECORD_TPCC_DELIV_DEF
txnDeliRec;

DWORD
index;

HANDLE
handles[2];

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

assert(txnDeliLog != NULL);

try
{
    if (Reg.eDB_Protocol
== ODBC)
    {
        if
(Reg.dwConnectDelay > 0)
        {
            // Synchronize connect (for VIA)
            //

            EnterCriticalSection(&hConnectCriticalSection);

            Sleep(Reg.dwConnectDelay);

            pTxn = pCTPCC_ODBC_new(
Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword,

            Reg.szMyComputerName,
Reg.szDbName,

            Reg.szSPPrefix,
Reg.bCallNoDuplicatesNewOrder );

            LeaveCriticalSection(&hConnectCriticalSection);
        }
        else
        {
            if
(Reg.eDB_Protocol == DBLIB)

            pTxn = pCTPCC_DBLIB_new(
Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName,
Reg.szDbName );
        }
        pDeliveryData =
pTxn->BuffAddr_Delivery();
    }
}

```

```

catch (CBaseErr *e)
    char szTmp[1024];
    wsprintf( szTmp,
"Error in Delivery Txn thread. Could not connect
to database. "

    "%s. Server=%s, User=%s,
Password=%s, Database=%s",

    e->ErrorText(), Reg.szDbServer,
Reg.szDbUser, Reg.szDbPassword,
Reg.szDbName );

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

while (TRUE)
{
    try
    {
        //while
        //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(&DelBuffCritical
Section);

        delivery =
*(pDelBuff+dwDelBuffBusyIndex);

        dwDelBuffFreeCount++;
        dwDelBuffBusyIndex++;
    }
}

```

```

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

    LeaveCriticalSection(&DelBuffCritical
Section);

    pDeliveryData->w_id =
delivery.w_id;

    pDeliveryData->o_carrier_id =
delivery.o_carrier_id;

    txnDeliRec.w_id = pDeliveryData-
>w_id;

    txnDeliRec.o_carrier_id =
pDeliveryData->o_carrier_id;

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

    GetLocalTime( &trans_start );

    pTxn->Delivery();

    GetLocalTime( &trans_end );

    //log txn

    txnDeliRec.TxnStatus =
ERR_SUCCESS;

    for (int i=0; i<10; i++)

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

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

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

    if (txnDeliLog != NULL)

        txnDeliLog-
>WriteToLog(&txnDeliRec);
    }
}
catch (CBaseErr *e)
{
    char
szTmp[1024];
    wsprintf(
szTmp, "%s Error (code %d) in Delivery Txn
thread. %s",

    e->ErrorTypeStr(),
e->ErrorNum(), e->ErrorText() );

    WriteMessageToEventLog( szTmp );
}

```

```

// log
the error txn
    txnDeliRec.TxnStatus = e-
>ErrorType();
    if
    (txnDeliLog != NULL)
        txnDeliLog-
>WriteToLog(&txnDeliRec);
        delete e;
    }
    catch (...)
    {
        //
        unhandled exception; shouldn't happen; not
        much we can do...
        WriteMessageToEventLog(TEXT("U
        nhandled exception caught in
        DeliveryWorkerThread.));
    }
}
ErrorExit:
    if (Reg.dwConnectDelay > 0)
    {
        // Synchronize
        disconnect (for VIA)
        //
        EnterCriticalSection(&hConnectCriti
        calSection);
        Sleep(Reg.dwConnectDelay);
        delete pTxn;
        LeaveCriticalSection(&hConnectCriti
        calSection);
    }
}
_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(&DelBuffCritical
    Section);
    if (dwDelBuffFreeCount > 0)
    {
        bError = FALSE;

```

```

    (pDelBuff+dwDelBuffFreeIndex)-
>w_id = w_id;
    (pDelBuff+dwDelBuffFreeIndex)-
>o_carrier_id= o_carrier_id;
    GetLocalTime(&(pDelBuff+dwDelBu
    ffFreeIndex)->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(&DelBuffCritical
        Section);
        if (!bError)
            // increment worker
            semaphore to wake up a worker thread
            ReleaseSemaphore(
            hWorkerSemaphore, 1, NULL );
        return bError;
    }
}
/* FUNCTION: ProcessQueryString
*
* PURPOSE: This function extracts the relevent
information out of the http command passed in
from
the
browser.
*
* COMMENTS:      If this is the initial
connection i.e. client is at welcome screen then
there will not be a terminal id or
current form id. If this is the case
then the pTermid and pFormid
return values are undefined.
*/
void
ProcessQueryString(EXTENSION_CONTROL_BLO
CK *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-Levels..", "Submit",
        "Menu", "Clear", "Stats", ""
    };
    *pCmd = 0;
    // default is the login screen
    *pTermId = 0;
    // if no params (i.e., empty query
    string), then return login screen
    if (strlen(pECB->lpszQueryString)
    == 0)
        return;
    // parse FORMID, TERMID, and
    SYNCID
    *pFormId = GetIntKeyValue(&ptr,
    "FORMID", NO_ERR, NO_ERR);
    *pTermId = GetIntKeyValue(&ptr,
    "TERMID", NO_ERR, NO_ERR);
    *pSyncId = GetIntKeyValue(&ptr,
    "SYNCID", NO_ERR, NO_ERR);
    // parse CMD
    GetKeyValue(&ptr, "CMD", szBuffer,
    sizeof(szBuffer), ERR_COMMAND_UNDEFINED);
    // see which command it matches
    for(i=0; ; i++)
    {
        if (szCmds[i][0] == 0)
            // no
            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=\"TERMINID\"
        VALUE=\"0\">"

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

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

    sprintf( szTmp,
        "Configuration Settings: <BR><font
        face=\"Courier New\" color=\"blue\"><PRE>"

        "Txn Monitor
        = <B>%s</B><BR>"

        "Database protocol
        = <B>%s</B><BR>"

        "Max Connections
        = <B>%d</B><BR>"

        "# of Delivery
        Threads = <B>%d</B><BR>"

        "Max Pending
        Deliveries = <B>%d</B><BR>"

        ,
        szTxnMonNames[Reg.eTxnMon],
        szDBNames[Reg.eDB_Protocol],

        Reg.dwMaxConnections,
        dwNumDeliveryThreads, dwDelBuffSize );
    strcat( szBuffer, szTmp);

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

```

```

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

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

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

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

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

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

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

                "</PRE></font>"

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

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

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

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

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

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

                "</PRE></font>"

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

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

```

```

        "<font face=\"Courier
        New\" color=\"blue\"><PRE><font color=\"blue\">
        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
            CWBCLNT_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, Reg.szSPPrefix,

Reg.bCallNoDuplicatesNewOrder );

        else if
(Reg.eDB_Protocol == DBLIB)

            Term.pClientData[iNewTerm].pTxn
= pCTPCC_DBLIB_new( szServer, szUser,
szPassword, szMyComputerName, szDatabase );
    }
    catch (...)
    {

```

```

TermDelete(iNewTerm);
// 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(&TermCriticalSe
ction);

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

            iTotal++;
    }

    LeaveCriticalSection(&TermCriticalS
ection);

    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_RAN
GE,

        "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_KE
Y,

        "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_EXCEED
ED,

        "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_INV
ALID, "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
ID, "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_WITHOU
T_SUPPW, "New Order Item_Id
field entered without a corresponding Supp_W."
    },
    {
      ERR_NEWORDER_MISSING_IID_KE
Y, "New
Order missing Item Id key \"IID*\"."
    },
    {
      ERR_NEWORDER_MISSING_QTY_K
EY, "New
Order Missing Qty key \"Qty##*\"."
    },
    {
      ERR_NEWORDER_MISSING_SUPPW
_KEY, "New
Order missing Supp_W key \"SP##*\"."
    },
    {
      ERR_NEWORDER_NOITEMS_ENTER
ED, "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_INVALI
D, "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_L
ONG, "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_THRE
SHOLD_KEY, "Stock Level; missing Threshold key
\"TT*\".",
    },
    {
        ERR_STOCKLEVEL_THRESHOLD_IN
VALID, "Stock Level;
Threshold value must be in the range = 1 - 99."
    },
    {
        ERR_STOCKLEVEL_THRESHOLD_RA
NGE, "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;
    }
    }
    }
    if (m_szTextDetail)
        strcat( szTmp,
m_szTextDetail );
    if (m_SystemErr)
        sprintf(
szTmp+strlen(szTmp), " Error=%d",
m_SystemErr);

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

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

```

```

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

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

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

    *pQueryString = ptr;
    return;

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

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

```



```

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

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

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

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

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

    *pQueryString = ptr;
    return atoi(ptr0);

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

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

void TermInit(void)
{

```

```

EnterCriticalSection(&TermCriticalSe
ction);
    Term.iMasterSyncId = 1;
    Term.iNumEntries =
Reg.dwMaxConnections+1;

    Term.pClientData = NULL;

    Term.pClientData =
(PCLIENTDATA)malloc(Term.iNumEntries *
sizeof(CLIENTDATA));
    if (Term.pClientData == NULL)
    {
        LeaveCriticalSection(&TermCriticalS
ection);
        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(&TermCriticalS
ection);
}

/* 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(&TermCriticalSe
ction);

    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(&TermCriticalS
ection);
}

/* 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(&TermCriticalSe
ction);
    if (Term.iFreeList != 0)
    {
        // position is available
iNewTerm =
Term.iFreeList;
        Term.iFreeList =
Term.pClientData[iNewTerm].iNextFree;

        Term.pClientData[iNewTerm].iNext
Free = -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(&TermCriticalS
ection);

```

```

throw
new CWEBCLNT_ERR(
ERR_MAX_CONNECTIONS_EXCEEDED );
}
}
Term.pClientData[iNewTerm].iTickCount = GetTickCount();
Term.pClientData[iNewTerm].iSyncId = Term.iMasterSyncId++;
Term.pClientData[iNewTerm].pTxn = NULL;

LeaveCriticalSection(&TermCriticalSection);
return iNewTerm;
}

/* FUNCTION: TermDelete
*
* PURPOSE: This function makes a terminal entry in the Term array available for reuse.
*
* ARGUMENTS:      int
                  id
                  Terminal id of client exiting
*/
void TermDelete(int id)
{
    if ( id > 0 && id <
Term.iNumEntries )
    {
        delete
Term.pClientData[id].pTxn;

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

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

        LeaveCriticalSection(&TermCriticalSection);
    }

/* FUNCTION: MakeErrorForm
*/
void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int iErrorNum, int iTermId, int iSyncId, char *szErrorText, char *szBuffer )
{
    wsprintf(szBuffer,

"<HTML><HEAD><TITLE>TPC-C Error</TITLE></HEAD><BODY>"
"<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
"<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">"
"<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
"<INPUT TYPE=\"hidden\" NAME=\"TERMINID\" VALUE=\"%d\">"
"<INPUT TYPE=\"hidden\" NAME=\"SYNCDID\" 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=\"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=\"SYNCDID\" VALUE=\"%d\">"
"<PRE><font face=\"Courier\"> Stock-Level<BR> Warehouse: %6.6d District: %2.2d<BR> <BR>,"
STOCK_LEVEL_FORM, iTermId, Term.pClientData[iTermId].iSyncId, Term.pClientData[iTermId].w_id, Term.pClientData[iTermId].d_id);
    if ( bInput )
    {
        strcpy(szForm+c,

```

```

"<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
"<INPUT TYPE=\"hidden\" NAME=\"TERMINID\" VALUE=\"%d\">"
"<INPUT TYPE=\"hidden\" NAME=\"SYNCDID\" 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>"
, iTermId, iSyncId, iErrorNum, 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=\"SYNCDID\" VALUE=\"%d\">"
"<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
"<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Payment..\">"

```

```

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

/* FUNCTION: MakeStockLevelForm
*
* PURPOSE: This function constructs the Stock Level HTML page.
*
* COMMENTS:      The internal client buffer is created when the terminal id is assigned and should not be freed except when the client terminal id is no longer needed.
*/
void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA *pStockLevelData, BOOL bInput, char *szForm)
{
    int c;

    c = wsprintf(szForm,

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

```

```

"Stock
Level Threshold: <INPUT NAME="TT*"
SIZE=2><BR> <BR>"
"low
stock: </font><BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR>"
" <BR>
<BR> <BR> <BR> <BR> <BR>
<BR></PRE><HR>"
" <INPUT
TYPE="submit" NAME="CMD"
VALUE="Process">"
" <INPUT
TYPE="submit" NAME="CMD"
VALUE="Menu">"
" </FORM></HTML>" );
}
else
{
    wsprintf(szForm+c,
"Stock
Level Threshold: %2.2d<BR> <BR>"
"low
stock: %3.3d</font> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR>"
" <BR>
<BR> <BR> <BR> <BR> <BR> <BR>
<BR></PRE><HR>"
" <INPUT
TYPE="submit" NAME="CMD"
VALUE="..NewOrder..">"
" <INPUT
TYPE="submit" NAME="CMD"
VALUE="..Payment..">"
" <INPUT
TYPE="submit" NAME="CMD"
VALUE="..Delivery..">"
" <INPUT
TYPE="submit" NAME="CMD"
VALUE="..Order-Status..">"
" <INPUT
TYPE="submit" NAME="CMD"
VALUE="..Stock-Level..">"
" <INPUT
TYPE="submit" NAME="CMD"
VALUE="..Exit..">"
" </FORM></HTML>"
pStockLevelData->threshold, pStockLevelData-
>low_stock);
}
}
/* FUNCTION: MakeNewOrderForm
*
* COMMENTS: The internal client
buffer is created when the terminal id is assigned
and should not
*
be freed except when the client
terminal id is no longer needed.
*/
void MakeNewOrderForm(int iTermId,
NEW_ORDER_DATA *pNewOrderData, BOOL
bInput, char *szForm)
{
    int i, c;
    BOOL bValid;
    static char szBR[] = " <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR>";

```

```

if (bInput) assert(
pNewOrderData->exec_status_code == eOK ||
pNewOrderData->exec_status_code ==
eInvalidItem );
bValid = (bInput ||
(pNewOrderData->exec_status_code == eOK));
c = wsprintf(szForm,
" <HTML><HEAD><TITLE>TPC-C
New Order</TITLE></HEAD><BODY>"
" <FORM
ACTION="tpcc.dll" METHOD="GET">"
" <INPUT
TYPE="hidden" NAME="STATUSID"
VALUE="%d">"
" <INPUT
TYPE="hidden" NAME="ERROR"
VALUE="0">"
" <INPUT
TYPE="hidden" NAME="FORMID"
VALUE="%d">"
" <INPUT
TYPE="hidden" NAME="TERMINID"
VALUE="%d">"
" <INPUT
TYPE="hidden" NAME="SYNCD"
VALUE="%d">"
" <PRE><font
face="Courier"> New
Order<BR>"
, bValid ? 0 :
ERR_BAD_ITEM_ID, NEW_ORDER_FORM,
iTermId, Term.pClientData[iTermId].iSynCd);
if ( bInput )
{
    c +=
wsprintf(szForm+c, "Warehouse: %6.6d ",
Term.pClientData[iTermId].w_id );
strcpy (szForm+c,
"District:
<INPUT NAME="DID*" SIZE=1>
Date:<BR>"
"Customer: <INPUT
NAME="CID*" SIZE=4> Name:
Credit: %Disc:<BR>"
"Order
W_tax:
Number: Number of Lines:
D_tax:<BR> <BR>"
"
Supp_W Item_Id Item Name Qty
Stock B/G Price Amount<BR>"
"
<INPUT NAME="SP00*" SIZE=4> <INPUT
NAME="IID00*" SIZE=6>
<INPUT NAME="Qty00*" SIZE=1><BR>"
"
<INPUT NAME="SP01*" SIZE=4> <INPUT
NAME="IID01*" SIZE=6>
<INPUT NAME="Qty01*" SIZE=1><BR>"
"
<INPUT NAME="SP02*" SIZE=4> <INPUT
NAME="IID02*" SIZE=6>
<INPUT NAME="Qty02*" SIZE=1><BR>"
"
<INPUT NAME="SP03*" SIZE=4> <INPUT
NAME="IID03*" SIZE=6>
<INPUT NAME="Qty03*" SIZE=1><BR>"

```

```

"
<INPUT NAME="SP04*" SIZE=4> <INPUT
NAME="IID04*" SIZE=6>
<INPUT NAME="Qty04*" SIZE=1><BR>"
<INPUT NAME="SP05*" SIZE=4> <INPUT
NAME="IID05*" SIZE=6>
<INPUT NAME="Qty05*" SIZE=1><BR>"
"
<INPUT NAME="SP06*" SIZE=4> <INPUT
NAME="IID06*" SIZE=6>
<INPUT NAME="Qty06*" SIZE=1><BR>"
"
<INPUT NAME="SP07*" SIZE=4> <INPUT
NAME="IID07*" SIZE=6>
<INPUT NAME="Qty07*" SIZE=1><BR>"
"
<INPUT NAME="SP08*" SIZE=4> <INPUT
NAME="IID08*" SIZE=6>
<INPUT NAME="Qty08*" SIZE=1><BR>"
"
<INPUT NAME="SP09*" SIZE=4> <INPUT
NAME="IID09*" SIZE=6>
<INPUT NAME="Qty09*" SIZE=1><BR>"
"
<INPUT NAME="SP10*" SIZE=4> <INPUT
NAME="IID10*" SIZE=6>
<INPUT NAME="Qty10*" SIZE=1><BR>"
"
<INPUT NAME="SP11*" SIZE=4> <INPUT
NAME="IID11*" SIZE=6>
<INPUT NAME="Qty11*" SIZE=1><BR>"
"
<INPUT NAME="SP12*" SIZE=4> <INPUT
NAME="IID12*" SIZE=6>
<INPUT NAME="Qty12*" SIZE=1><BR>"
"
<INPUT NAME="SP13*" SIZE=4> <INPUT
NAME="IID13*" SIZE=6>
<INPUT NAME="Qty13*" SIZE=1><BR>"
"
<INPUT NAME="SP14*" SIZE=4> <INPUT
NAME="IID14*" SIZE=6>
<INPUT NAME="Qty14*" SIZE=1><BR>"
"
"Execution Status:
Total:<BR>"
" </font></PRE><HR>"
" <INPUT
TYPE="submit" NAME="CMD"
VALUE="Process">"
" <INPUT
TYPE="submit" NAME="CMD"
VALUE="Menu">"
" </FORM></HTML>"
);
}
else
{
    c +=
wsprintf(szForm+c, "Warehouse: %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 +=
    sprintf(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,
                100.0 * pNewOrderData->w_tax,
                100.0 * pNewOrderData->d_tax);
            for(i=0;
i<pNewOrderData->o_ol_cnt; i++)
            {
                c += sprintf(szForm+c, "%6.6d
%6.6d %-24s %2.2d %3.3d %1.1s
%5.2f %5.2f <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 +=
        sprintf(szForm+c,
            "%Disc: <BR>"
            "Order Number: %8.8d Number of
Lines: W_tax: D_tax: <BR> <BR>"
            " Supp_W Item_Id Item Name
Qty Stock B/G Price Amount<BR>"
            , pNewOrderData->o_id);
        i = 0;
    }
    strncpy( szForm+c,
szBR, (15-i)*5 );
    c += (15-i)*5;
    if ( bValid )
    c +=
    sprintf(szForm+c, "Execution Status: Transaction
committed. Total: $%8.2f ",
        pNewOrderData->total_amount);
    else
    c +=
    sprintf(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(szForm,
        "<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD><BODY>"
        "<FORM
ACTION=\"tpcc.dll\" METHOD=\"GET\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"0\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"TERMINID\"
VALUE=\"%d\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"SYNCID\"
VALUE=\"%d\">"
        "<PRE><font
face=\"Courier\">
Payment<BR>"
        "Date: "
        , PAYMENT_FORM,
        iTermId, Term.pClientData[iTermId].iSyncid);
    if ( !bInput )
    {
        c +=
        sprintf(szForm+c, "%2.2d-%2.2d-%4.4d
%2.2d:%2.2d:%2.2d",
            pPaymentData->h_date.day,
            pPaymentData->h_date.month,
            pPaymentData->h_date.year,
            pPaymentData->h_date.hour,
            pPaymentData->h_date.minute,
            pPaymentData->h_date.second);
    }
    if ( bInput )
    {
        c +=
        sprintf(szForm+c,
            "<BR>Warehouse: %6.6d"
            "
            District: <INPUT NAME=\"DID\" SIZE=1><BR>
<BR> <BR> <BR> <BR>"
            "Customer: <INPUT
NAME=\"CID\" SIZE=4>"

```



```

    {
        strcpy(szForm+c,
            "District:
<INPUT NAME=\"DID*\" SIZE=1><BR>\"
            "Customer: <INPUT
NAME=\"CID*\" SIZE=4> Name:
<INPUT NAME=\"CLT*\" SIZE=23><BR>\"
            "Cust-
Balance:<BR> <BR>\"
            "Order-
Number:      Entry-Date:
Carrier-Number:<BR>\"
            "Supply-
W Item-Id Qty Amount Delivery-
Date<BR> <BR> <BR> <BR> <BR>\"
            " <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR></font></PRE>\"
            " <HR><INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\"><INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Menu\">\"
            " </BODY></FORM></HTML>\" );
        }
        else
        {
            c +=
            sprintf(szForm+c,
                "District:
                %2.2d<BR>\"
                "Customer: %4.4d Name: %6.16s
                %-2s %6.16s<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-
        >o_entry_d.second,
        pOrderStatusData->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);
            strcpy( 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> <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> <BR>
                </font></PRE>\"
                " <HR><INPUT TYPE=\"submit\"
                NAME=\"CMD\" VALUE=\"..NewOrder..\">\"

```

```

"TYPE=\submit\ NAME=\CMD\
VALUE=\..Payment..\>"
"TYPE=\submit\ NAME=\CMD\
VALUE=\..Delivery..\>"
"TYPE=\submit\ NAME=\CMD\
VALUE=\..Order-Status..\>"
"TYPE=\submit\ NAME=\CMD\
VALUE=\..Stock-Level..\>"
"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_B
LOCK *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_B
LOCK *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)
ORDER_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_B
LOCK *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 inetrv.
*
*
* int
*
* iTermId client browser
terminal id
*
*/

void
ProcessStockLevelForm(EXTENSION_CONTROL_B
LOCK *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
                    lpszQueryString
                    client browser http
command string
*
                    PAYMENT_DATA
                    *pPaymentData
                    pointer to payment data structure
*/

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

    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_CUSTOMER_INVALID);
        pPaymentData->d_id =
GetIntKeyValue(&ptr, "DID*",
ERR_PAYMENT_MISSING_DID_KEY,
ERR_PAYMENT_DISTRICT_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(szTmp) >
LAST_NAME_LEN )
                throw
new CWEBCLNT_ERR(
ERR_PAYMENT_LAST_NAME_TO_LONG );

            strcpy(pPaymentData-
>c_last, szTmp);
            // pad with spaces so
that the client layer doesn't have to do it
            // before passing
parameters to stored procedure
            iLen =
strlen(pPaymentData->c_last);
            memset(pPaymentData->c_last +
iLen, ' ', LAST_NAME_LEN - iLen);
            pPaymentData-
>c_last[LAST_NAME_LEN] = 0;
        }
        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;
    int         iLen;

    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 );

        _strup( szTmp );
        if ( strlen(szTmp) >
LAST_NAME_LEN )
            throw
new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CLT_RANGE );

        strcpy(pOrderStatusData->c_last,
szTmp);
        // pad with spaces so
that the client layer doesn't have to do it
        // before passing
parameters to stored procedure
        iLen =
strlen(pOrderStatusData->c_last);
        memset(pOrderStatusData->c_last
+ iLen, ' ', LAST_NAME_LEN - iLen);
        pOrderStatusData-
>c_last[LAST_NAME_LEN] = 0;
    }
    else
    {
        // parse customer id
and verify that last name was NOT entered
        if ( !IsNumeric(szTmp)
)
            throw
new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CID_INVALID );
        pOrderStatusData-
>c_id = atoi(szTmp);
        GetKeyValue(&ptr,
"CLT*", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CLT_KEY);
        if ( szTmp[0] != 0 )
            throw
new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CID_AND_CLT );
    }
}

```

```

/* FUNCTION: BOOL IsNumeric(char *ptr)
*
* PURPOSE: This function determines if a string
is numeric. It fails if any characters other
* than
numeric and null terminator are present.
*
* ARGUMENTS:      char      pointer
                 *ptr
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      pointer
                 *ptr
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;
}

/* 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\txnolog\include\rtetime.h"
#include
"..\..\common\txnolog\include\spinlock.h"
#include "..\..\common\txnolog\include\txnolog.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_COMPUT
ERNAME_LENGTH+1];

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

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

static CRITICAL_SECTION
TermCriticalSection;

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

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

// For deferred Delivery txns:
CTxnLog
*txnDelilog = NULL;
//used
to log delivery transaction information

HANDLE
hWorkerSemaphore
= INVALID_HANDLE_VALUE;

```

```

HANDLE
    hDoneEvent
    =
INVALID_HANDLE_VALUE;
HANDLE
    *pDelihandles
    = NULL;

// configuration settings from registry
TPCCREGISTRYDATA Reg;

DWORD
    dwNumDeliveryThreads = 4;
CRITICAL_SECTION
    DelBuffCriticalSection;
//critical section for delivery
transactions cache
DELIVERY_TRANSACTION *pDelBuff
    = NULL;
DWORD
    dwDelBuffSize
    = 100;
// size of circular buffer for delivery
txns
DWORD
    dwDelBuffFreeCount;

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

// Critical section to synchronize connection open
and close.
//
CRITICAL_SECTION hConnectCriticalSection;

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

/* FUNCTION: DllMain
 *
 * PURPOSE: This function is the entry point for
the DLL. This implementation is based on the
 * fact that
DLL_PROCESS_ATTACH is only called from the
inet service once.
 *
 * ARGUMENTS: HANDLE hModule
    module handle
 *
    DWORD ul_reason_for_call
    reason for call
 *
    LPVOID lpReserved
    reserved for future
use
 *
 * RETURNS: BOOL FALSE
    errors 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(szMyComputerN
                    ame, &dwSize);

                szMyComputerName[dwSize] = 0;
            }

            DisableThreadLibraryCalls((HMODU
                LE)hModule);

            InitializeCriticalSection(&TermCritica
                lSection);

            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 bxn monitor

                if (Reg.eTxnMon == TUXEDO)
                {
                    strcpy( szDllName,
                        Reg.szPath );

                    strcat( szDllName,
                        "tpcc_tuxedo.dll");
                }
            }
        }
    }

```

```

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

                // get function pointer
                to wrapper for class constructor

                pCTPCC_TUXEDO_new =
                (TYPE_CTPCC_TUXEDO*)
                GetProcAddress(hLibInstanceTm,"CTPCC_TUXED
                    O_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_ENCIN
                            A_new");

                        pCTPCC_ENCINA_post_init =
                        (TYPE_CTPCC_ENCINA*)
                        GetProcAddress(hLibInstanceTm,"CTPCC_ENCIN
                            A_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_n
ew");

    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)

        {

            // Initialize delivery
delay critical section

            //

```

```

InitializeCriticalSection(&hConnectC
riticalSection);

        // for deferred
delivery txns:

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

        InitializeCriticalSection(&DelBuffCrite
calSection);

            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-
%2.2ds%2.2dms.log",

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

            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(&DelBuffCriti
            calSection);

```

```

        // Delete delivery
        delay critical section
        //

        DeleteCriticalSection(&hConnectCriti
        calSection);
    }

    DeleteCriticalSection(&TermCriticalS
    ection);

    if (hLibInstanceTm != NULL)

        FreeLibrary(
        hLibInstanceTm );

        hLibInstanceTm = NULL;

    if (hLibInstanceDb != NULL)

        FreeLibrary(
        hLibInstanceDb );

        hLibInstanceDb = NULL;

        Sleep(500);

        break;

        default:

            /* nothing */;
        }
    }
    catch (CBaseErr *e)
    {
        TCHAR szMsg[256];

        _sntprintf(szMsg,
        sizeof(szMsg), "%s error, code %d: %s",
        e->ErrorTypeStr(), e->ErrorNum(),
        e->ErrorText());

        WriteMessageToEventLog( szMsg );
        delete e;

        TerminateExtension(0);
        return FALSE;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("U
        nhandled 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];

                sprintf( 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:
    FormId )
    switch(
    {
    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;
    case 2:
        // new-
        order selected from menu; display new-order
        input form

```

```

        MakeNewOrderForm(TermId, NULL,
INPUT_FORM, szBuffer);
        break;
    case 3:
        //
        payment selected from menu; display payment
        input form
        MakePaymentForm(TermId, NULL,
INPUT_FORM, szBuffer);
        break;
    case 4:
        //
        delivery selected from menu; display delivery
        input form
        MakeDeliveryForm(TermId, NULL,
INPUT_FORM, szBuffer);
        break;
    case 5:
        // order-
        status selected from menu; display order-status
        input form
        MakeOrderStatusForm(TermId,
NULL, INPUT_FORM, szBuffer);
        break;
    case 6:
        // stock-
        level selected from menu; display stock-level
        input form
        MakeStockLevelForm(TermId,
NULL, INPUT_FORM, szBuffer);
        break;
    case 7:
        //
        ExitCmd
        TermDelete(TermId);
        WelcomeForm(pECB, szBuffer);
        break;
    case 8:
        SubmitCmd(pECB, szBuffer);
        break;
    case 9:
        // menu
        MakeMainMenuForm(TermId,
Term.pClientData[TermId].iSyncId, szBuffer);
        break;
    case 10:
        //
        CMD=Clear
        // resets
        all connections; should only be used when no
        other connections are active
        TermDeleteAll();

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

```

```

        catch (CBaseErr *e)
        {
            ErrorForm( pECB, e-
>ErrorType(), e->ErrorNum(), TermId, iSyncId,
e->ErrorText(), szBuffer );
            delete e;
        }
        catch (...)
        {
            ErrorForm( pECB,
ERR_TYPE_WEBDLL, 0, TermId, iSyncId, "Error:
Unhandled exception in Web Client.", szBuffer );
        }

#ifdef ICECAP
        StopCAP();
#endif

        lpbSize = strlen(szBuffer);
        sprintf(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"));

    _sprintf(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)
        {
            if
(Reg.dwConnectDelay > 0)
            {
                // Synchronize connect (for VIA)
                //
                EnterCriticalSection(&hConnectCritic
alSection);

                Sleep(Reg.dwConnectDelay);

                pTxn = pCTPCC_ODBC_new(
Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword,

```

```

                szMyComputerName,
Reg.szDbName,

                Reg.szSPPrefix,
Reg.bCallNoDuplicatesNewOrder );

                LeaveCriticalSection(&hConnectCritic
alSection);
            }
        }
        else
        {
            if
(Reg.eDB_Protocol == DBLIB)

                pTxn = pCTPCC_DBLIB_new(
Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName,
Reg.szDbName );

                pDeliveryData =
pTxn->BuffAddr_Delivery();
            }
            catch (CBaseErr *e)
            {
                char szTmp[1024];
                sprintf( 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("U
nhanded exception caught in
DeliveryWorkerThread.));
                goto ErrorExit;
            }
            while (TRUE)
            {
                try
                {
                    //while
delivery thread running, i.e. user has not
requested termination

                    while
(TRUE)
                    {
                        // need to wait for multiple objects:
program exit or worker semaphore;

                        handles[0] = hDoneEvent;
                        handles[1] = hWorkerSemaphore;

```

```

        index = WaitForMultipleObjects( 2,
&handles[0], FALSE, INFINITE );

        if (index == WAIT_OBJECT_0)

            goto ErrorExit;

        ZeroMemory(&txnDeliRec,
sizeof(txnDeliRec));

        txnDeliRec.TxnType =
TXN_REC_TYPE_TPCC_DELIV_DEF;

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

        EnterCriticalSection(&DelBuffCritical
Section);

        delivery =
*(pDelBuff+dwDelBuffBusyIndex);

        dwDelBuffFreeCount++;

        dwDelBuffBusyIndex++;

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

            dwDelBuffBusyIndex
= 0;

        LeaveCriticalSection(&DelBuffCritical
Section);

        pDeliveryData->w_id =
delivery.w_id;

        pDeliveryData->o_carrier_id =
delivery.o_carrier_id;

        txnDeliRec.w_id = pDeliveryData-
>w_id;

        txnDeliRec.o_carrier_id =
pDeliveryData->o_carrier_id;

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

        GetLocalTime( &trans_start );

        pTxn->Delivery();

        GetLocalTime( &trans_end );

        //log txn

        txnDeliRec.TxnStatus =
ERR_SUCCESS;

        for (int i=0; i<10; i++)

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

```

```

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

        if (txnDelilog != NULL)

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

            wsprintf(
szTmp, "%s Error (code %d) in Delivery Txn
thread. %s",
e->ErrorTypeStr(),
e->ErrorNum(), e->ErrorText() );

            WriteMessageToEventLog( szTmp );

            // log
the error txn

            txnDeliRec.TxnStatus = e-
>ErrorType();

            if
(txnDelilog != NULL)

                txnDelilog-
>WriteToLog(&txnDeliRec);

            delete e;
        }
        catch (...)
        {
            //
unhandled exception; shouldn't happen; not
much we can do...

            WriteMessageToEventLog(TEXT("U
nhandled exception caught in
DeliveryWorkerThread."));
        }
    }

ErrorExit:
    if (Reg.dwConnectDelay > 0)
    {
        // Synchronize
disconnect (for VIA)

        //

        EnterCriticalSection(&hConnectCriti
calSection);

        Sleep(Reg.dwConnectDelay);

        delete pTxn;

        LeaveCriticalSection(&hConnectCritic
alSection);
    }
}

```

```

    }

    _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(&DelBuffCritical
Section);
    if (dwDelBuffFreeCount > 0)
    {
        bError = FALSE;

        (pDelBuff+dwDelBuffFreeIndex)-
>w_id = w_id;

        (pDelBuff+dwDelBuffFreeIndex)-
>o_carrier_id= o_carrier_id;

        GetLocalTime(&(pDelBuff+dwDelBu
ffFreeIndex)->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(&DelBuffCritical
Section);

        if (!bError)

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

        return bError;
    }
}

/* FUNCTION: ProcessQueryString
*
* PURPOSE: This function extracts the relevent
information out of the http command passed in
from
*
                the
browser.

```



```

*
* COMMENTS:      If this is the initial
connection i.e. client is at welcome screen then
*
                there will not be a terminal id or
current form id. If this is the case
*
                then the pTermId and pFormId
return values are undefined.
*/

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

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

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

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

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

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

    // see which command it matches
for(i=0; ; i++)
    {
        if (szCmds[i][0] == 0)
            // no
            throw
more; no match; return error
        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>"

        "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>"

        "<B>%s</B><BR>" "Txn Monitor
        "<B>%s</B><BR>" "Database protocol
        "<B>%d</B><BR>" "Max Connections

```

```

        "# 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>"

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

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

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

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

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

        "</PRE></font>"

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

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

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

```

```

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

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

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

" </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, Reg.szSPPrefix,

Reg.bCallNoDuplicatesNewOrder );
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 iTTotal;

EnterCriticalSection(&TermCriticalSe
ction);

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

iTTotal++;
}

LeaveCriticalSection(&TermCriticalS
ection);

wsprintf( szBuffer,

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

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

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."
},
{
    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_INVALI
        D,
        "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_L
        ONG,
        "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_KE
        Y,
        "Payment missing Amount key
        \"HAM*\"."
    },
    {
        ERR_STOCKLEVEL_MISSING_THRE
        SHOLD_KEY, "Stock Level; missing Threshold key
        \"TT*\"."
    },
    {
        ERR_STOCKLEVEL_THRESHOLD_IN
        VALID,
        "Stock Level;
        Threshold value must be in the range = 1 - 99."
    },
    {
        ERR_STOCKLEVEL_THRESHOLD_RA
        NGE,
        "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)
    sprintf(
    szTmp+strlen(szTmp), " Error=%d",
    m_SystemErr );

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

/* FUNCTION: GetKeyValue
*
* PURPOSE: This function parses a http
* formatted string for specific key values.
*
* ARGUMENTS:      char
*                  *pQueryString
*                  http string from client browser
*
* char
* *pKey
* key value to look for
*
* char
* *pValue
* character array into which to place
key's value
*
* int
* iMax
* maximum length of
key value array.
*
* WEBERROR      err
*
* error value to throw

```

```

*
* RETURNS:      nothing.
*
* ERROR:       if (the pKey value is
not found) then
*
*              if (err == 0)
*
*                  return
(empty string)
*
*              else
*
*                  throw
CWEBCLNT_ERR(err)
*
* COMMENTS:    http keys are
formatted either KEY=value& or KEY=value\0.
This DLL formats
*
*              TPC-C input fields in such a manner
that the keys can be extracted in the
*
*              above manner.
*/

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

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

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

    *pQueryString = ptr;
    return;

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

/* FUNCTION: GetIntKeyValue
*
* PURPOSE: This function parses a http
formatted string for a specific key value.
*
* ARGUMENTS:  char
              *pQueryString
              http string from client browser
*
              char
              *pKey
              key value to look for
*
              WEBERROR
              NoKeyErr      error
value to throw if key not found

```

```

*
*              WEBERROR
              NotIntErr      error
value to throw if value not numeric
integer
* RETURNS:
*
* 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(&TermCriticalSe
ction);

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

    Term.pClientData = NULL;
    Term.pClientData =
(PCLIENTDATA)malloc(Term.iNumEntries *
sizeof(CLIENTDATA));
    if (Term.pClientData == NULL)
    {
        LeaveCriticalSection(&TermCriticalS
ection);
        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(&TermCriticalS
ection);
}

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

```

```

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

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

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

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

    free(Term.pClientData);
    Term.pClientData
= NULL;

    LeaveCriticalSection(&TermCriticalSection);
}

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

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

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

    EnterCriticalSection(&TermCriticalSection);
    if (Term.iFreeList != 0)
    {
        // position is available
        iNewTerm =

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

        Term.pClientData[iNewTerm].iNextFree = -1; // indicates this position is in use
    }
    else
    {
        // no open slots, so
        find the slot that hasn't been used in the longest
        time and reuse it
        for(iNewTerm=1, i=1,
iTickCount=0x7FFFFFFF;
i<Reg.dwMaxConnections; i++)
        {
            if
(iTickCount > Term.pClientData[i].iTickCount)
            {

```

```

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

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

    LeaveCriticalSection(&TermCriticalSection);
    return iNewTerm;
}

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

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

        // put onto free list

        EnterCriticalSection(&TermCriticalSection);

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

        LeaveCriticalSection(&TermCriticalSection);
    }
}

```

```

/* FUNCTION: MakeErrorForm
*/

void ErrorForm(EXTENSION_CONTROL_BLOCK
*pECB, int iType, int iErrorNum, int iTermId, int
iSyncId, char *szErrorText, char *szBuffer )
{
    wsprintf(szBuffer,

    "<HTML><HEAD><TITLE>TPC-C
Error</TITLE></HEAD><BODY>"
    "<FORM
ACTION=\"tpcc.dll\" METHOD=\"GET\">"
    "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"%d\">"
    "<INPUT
TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"%d\">"
    "<INPUT
TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"
    "<INPUT
TYPE=\"hidden\" NAME=\"TERMINID\"
VALUE=\"%d\">"
    "<INPUT
TYPE=\"hidden\" NAME=\"SYNCDID\"
VALUE=\"%d\">"
    "<BOLD>An Error
Occurred</BOLD><BR><BR>"
    "%s"
    "<BR><BR><HR>"
    "<INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
    "<INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"
    "<INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"
    "<INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Order-Status..\">"
    "<INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Stock-Level..\">"
    "<INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Exit..\">"

    "</FORM></BODY></HTML>"
    , iType, iErrorNum,
MAIN_MENU_FORM, iTermId, iSyncId,
szErrorText );
}

/* FUNCTION: MakeMainMenuForm
*/

void MakeMainMenuForm(int iTermId, int
iSyncId, char *szForm)
{
    wsprintf(szForm,

    "<HTML><HEAD><TITLE>TPC-C
Main Menu</TITLE></HEAD><BODY>"
    "Select Desired
Transaction.<BR><HR>"
    "<FORM
ACTION=\"tpcc.dll\" METHOD=\"GET\">"

```

```

        "<INPUT
TYPE=\\"hidden\\" NAME=\\"STATUSID\\"
VALUE=\\"0\\">"
        "<INPUT
TYPE=\\"hidden\\" NAME=\\"ERROR\\"
VALUE=\\"0\\">"
        "<INPUT
TYPE=\\"hidden\\" NAME=\\"FORMID\\"
VALUE=\\"%d\\">"
        "<INPUT
TYPE=\\"hidden\\" NAME=\\"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)
{
    int        c;

    c = sprintf(szForm,

        "<HTML><HEAD><TITLE>TPC-C
Stock Level</TITLE></HEAD><FORM
ACTION=\\"tpcc.dll\\" METHOD=\\"GET\\">"
        "<INPUT
TYPE=\\"hidden\\" NAME=\\"STATUSID\\"
VALUE=\\"0\\">"
        "<INPUT
TYPE=\\"hidden\\" NAME=\\"ERROR\\"
VALUE=\\"0\\">"
        "<INPUT
TYPE=\\"hidden\\" NAME=\\"FORMID\\"
VALUE=\\"%d\\">"
        "<INPUT
TYPE=\\"hidden\\" NAME=\\"TERMINID\\"
VALUE=\\"%d\\">"
        "<INPUT
TYPE=\\"hidden\\" NAME=\\"SYNCID\\"
VALUE=\\"%d\\">"

```

```

        "<INPUT
TYPE=\\"hidden\\" NAME=\\"SYNCID\\"
VALUE=\\"%d\\">"
        "<PRE><font
face=\\"Courier\\">
                Stock
Level<BR>"
        "Warehouse: %6.6d
District: %2.2d<BR> <BR>",
        STOCK_LEVEL_FORM,
iTermId, Term.pClientData[iTermId].iSyncId,
        Term.pClientData[iTermId].w_id,
Term.pClientData[iTermId].d_id);

        if ( bInput )
        {
                strcpy(szForm+c,

                        "Stock
Level Threshold: <INPUT NAME=\\"TT*\\"
SIZE=2><BR> <BR>"
                        "low
stock: </font><BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR>"
                        " <BR>
<BR> <BR> <BR> <BR> <BR>
<BR></PRE><HR>"
                        "<INPUT
TYPE=\\"submit\\" NAME=\\"CMD\\"
VALUE=\\"Process\\>"
                        "<INPUT
TYPE=\\"submit\\" NAME=\\"CMD\\"
VALUE=\\"Menu\\>"

                "</FORM></HTML>" );
        }
        else
        {
                sprintf(szForm+c,

                        "Stock
Level Threshold: %2.2d<BR> <BR>"
                        "low
stock: %3.3d</font> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR>"
                        " <BR>
<BR> <BR> <BR> <BR> <BR> <BR>
<BR></PRE><HR>"
                        "<INPUT
TYPE=\\"submit\\" NAME=\\"CMD\\"
VALUE=\\"..NewOrder..\\>"
                        "<INPUT
TYPE=\\"submit\\" NAME=\\"CMD\\"
VALUE=\\"..Payment..\\>"
                        "<INPUT
TYPE=\\"submit\\" NAME=\\"CMD\\"
VALUE=\\"..Delivery..\\>"
                        "<INPUT
TYPE=\\"submit\\" NAME=\\"CMD\\"
VALUE=\\"..Order-Status..\\>"
                        "<INPUT
TYPE=\\"submit\\" NAME=\\"CMD\\"
VALUE=\\"..Stock-Level..\\>"
                        "<INPUT
TYPE=\\"submit\\" NAME=\\"CMD\\"
VALUE=\\"..Exit..\\>"

                "</FORM></HTML>"

                pStockLevelData->threshold, pStockLevelData-
>low_stock);
        }

/* FUNCTION: MakeNewOrderForm
*

```

```

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

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

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

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

    c = sprintf(szForm,

        "<HTML><HEAD><TITLE>TPC-C
New Order</TITLE></HEAD><BODY>"
        "<FORM
ACTION=\\"tpcc.dll\\" METHOD=\\"GET\\">"
        "<INPUT
TYPE=\\"hidden\\" NAME=\\"STATUSID\\"
VALUE=\\"%d\\">"
        "<INPUT
TYPE=\\"hidden\\" NAME=\\"ERROR\\"
VALUE=\\"0\\">"
        "<INPUT
TYPE=\\"hidden\\" NAME=\\"FORMID\\"
VALUE=\\"%d\\">"
        "<INPUT
TYPE=\\"hidden\\" NAME=\\"TERMINID\\"
VALUE=\\"%d\\">"
        "<INPUT
TYPE=\\"hidden\\" NAME=\\"SYNCID\\"
VALUE=\\"%d\\">"

        "<PRE><font
face=\\"Courier\\">
                New
Order<BR>"
        , bValid ? 0 :
ERR_BAD_ITEM_ID, NEW_ORDER_FORM,
iTermId, Term.pClientData[iTermId].iSyncId);

        if ( bInput )
        {
                c +=
sprintf(szForm+c, "Warehouse: %6.6d ",
Term.pClientData[iTermId].w_id);

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

                        "Customer: <INPUT
NAME=\\"CID*\\" SIZE=4> Name:
Credit: %Disc:<BR>"

                        "Order
Number:      Number of Lines:      W_tax:
D_tax:<BR> <BR>"

```

```

"
Supp_W Item_Id Item Name Qty
Stock B/G Price Amount<BR>"
"
<INPUT NAME="SP00*" SIZE=4> <INPUT
NAME="IID00*" SIZE=6>
<INPUT NAME="Qty00*" SIZE=1><BR>"
"
<INPUT NAME="SP01*" SIZE=4> <INPUT
NAME="IID01*" SIZE=6>
<INPUT NAME="Qty01*" SIZE=1><BR>"
"
<INPUT NAME="SP02*" SIZE=4> <INPUT
NAME="IID02*" SIZE=6>
<INPUT NAME="Qty02*" SIZE=1><BR>"
"
<INPUT NAME="SP03*" SIZE=4> <INPUT
NAME="IID03*" SIZE=6>
<INPUT NAME="Qty03*" SIZE=1><BR>"
"
<INPUT NAME="SP04*" SIZE=4> <INPUT
NAME="IID04*" SIZE=6>
<INPUT NAME="Qty04*" SIZE=1><BR>"
"
<INPUT NAME="SP05*" SIZE=4> <INPUT
NAME="IID05*" SIZE=6>
<INPUT NAME="Qty05*" SIZE=1><BR>"
"
<INPUT NAME="SP06*" SIZE=4> <INPUT
NAME="IID06*" SIZE=6>
<INPUT NAME="Qty06*" SIZE=1><BR>"
"
<INPUT NAME="SP07*" SIZE=4> <INPUT
NAME="IID07*" SIZE=6>
<INPUT NAME="Qty07*" SIZE=1><BR>"
"
<INPUT NAME="SP08*" SIZE=4> <INPUT
NAME="IID08*" SIZE=6>
<INPUT NAME="Qty08*" SIZE=1><BR>"
"
<INPUT NAME="SP09*" SIZE=4> <INPUT
NAME="IID09*" SIZE=6>
<INPUT NAME="Qty09*" SIZE=1><BR>"
"
<INPUT NAME="SP10*" SIZE=4> <INPUT
NAME="IID10*" SIZE=6>
<INPUT NAME="Qty10*" SIZE=1><BR>"
"
<INPUT NAME="SP11*" SIZE=4> <INPUT
NAME="IID11*" SIZE=6>
<INPUT NAME="Qty11*" SIZE=1><BR>"
"
<INPUT NAME="SP12*" SIZE=4> <INPUT
NAME="IID12*" SIZE=6>
<INPUT NAME="Qty12*" SIZE=1><BR>"
"
<INPUT NAME="SP13*" SIZE=4> <INPUT
NAME="IID13*" SIZE=6>
<INPUT NAME="Qty13*" SIZE=1><BR>"
"
<INPUT NAME="SP14*" SIZE=4> <INPUT
NAME="IID14*" SIZE=6>
<INPUT NAME="Qty14*" SIZE=1><BR>"
"
"Execution Status:
Total:<BR>"
"
</font></PRE><HR>"
" <INPUT
TYPE="submit" NAME="CMD"
VALUE="Process">"
" <INPUT
TYPE="submit" NAME="CMD"
VALUE="Menu">"

```

```

"</FORM></HTML>" );
}
else
{
    c +=
    sprintf(szForm+c, "Warehouse: %6.6d
District: %2.2d Date: ",
    pNewOrderData->w_id,
    pNewOrderData->d_id);
    if ( bValid )
    {
        c +=
        sprintf(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 +=
    sprintf(szForm+c, "<BR>Customer: %4.4d
Name: %-16s Credit: %-2s ",
    pNewOrderData->c_id,
    pNewOrderData->c_last, pNewOrderData-
>c_credit);
    if ( bValid )
    {
        c +=
        sprintf(szForm+c,
        "%%Disc: %5.2f <BR>"
        "Order
Number: %8.8d Number of Lines: %2.2d
W_tax: %5.2f D_tax: %5.2f <BR> <BR>"
        "
Supp_W Item_Id Item Name Qty
Stock B/G Price Amount<BR>",
        100.0*pNewOrderData-
>c_discount,
        pNewOrderData->o_id,
        pNewOrderData->o_ol_cnt,
        100.0 * pNewOrderData->w_tax,
        100.0 * pNewOrderData->d_tax);
        for(i=0;
i<pNewOrderData->o_ol_cnt; i++)
        {

```

```

    c += sprintf(szForm+c, ""%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 +=
        sprintf(szForm+c,
        "%Disc:<BR>"
        "Order Number: %8.8d Number of
Lines:
W_tax: D_tax:<BR> <BR>"
        "Supp_W Item_Id Item Name
Qty Stock B/G Price Amount<BR>"
        , pNewOrderData->o_id);
        i = 0;
        strncpy( szForm+c,
        szBR, (15-i)*5 );
        c += (15-i)*5;
        if ( bValid )
        {
            c +=
            sprintf(szForm+c, "Execution Status: Transaction
committed. Total: $%6.2f ",
            pNewOrderData->total_amount);
        }
        else
        {
            c +=
            sprintf(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(szForm,
" <HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD><BODY>"
" <FORM
ACTION="tpcc.dll" METHOD="GET">"
" <INPUT
TYPE="hidden" NAME="STATUSID"
VALUE="0">"
" <INPUT
TYPE="hidden" NAME="ERROR"
VALUE="0">"
" <INPUT
TYPE="hidden" NAME="FORMID"
VALUE="%">"
" <INPUT
TYPE="hidden" NAME="TERMIN"
VALUE="%">"
" <INPUT
TYPE="hidden" NAME="SYNCD"
VALUE="%">"
" <PRE><font
face="Courier">
Payment<BR>"
"Date: "
, PAYMENT_FORM,
iTermId, Term.pClientData[iTermId].iSyncd);
if ( !bInput )
{
c +=
sprintf(szForm+c, "%2.2d-%2.2d-%4.4d
%2.2d:%2.2d:%2.2d",
pPaymentData->h_date.day,
pPaymentData->h_date.month,
pPaymentData->h_date.year,
pPaymentData->h_date.hour,

```

```

pPaymentData->h_date.minute,
pPaymentData->h_date.second);
}
if ( bInput )
{
c +=
sprintf(szForm+c,
" <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(szForm+c,
" <BR>Warehouse: %6.6d
District: %2.2d<BR>"
"%-20s
%-20s<BR>"
"%-20s
%-20s<BR>"
"%-20s
%-2s %5.5s-%4.4s %-20s %-2s %5.5s-
%4.4s<BR> <BR>"
"Customer: %4.4d Cust-
Warehouse: %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, pPaymentData-
>c_credit
);
c += sprintf(szForm+c,
"%-20s
%%Disc: %5.2f<BR>",
pPaymentData->c_street_2,
100.0*pPaymentData->c_discount);
c +=
sprintf(szForm+c,
"%-20s %-2s %5.5s-%4.4s Phone: %6.6s-
%3.3s-%3.3s-%4.4s<BR> <BR>",
pPaymentData->c_city,
pPaymentData->c_state, pPaymentData->c_zip,
pPaymentData->c_zip+5,
pPaymentData->c_phone,
pPaymentData->c_phone+6, pPaymentData-
>c_phone+9, pPaymentData->c_phone+12 );
c +=
sprintf(szForm+c,
"Amount
Paid: $%7.2f New Cust-Balance:
$%14.2f<BR>"
"Credit
Limit: $%13.2f<BR> <BR>"
pPaymentData->h_amount, pPaymentData-
>c_balance
pPaymentData->c_credit_lim
);
if ( pPaymentData-
>c_credit[0] == 'B' && pPaymentData-
>c_credit[1] == 'C' )
c +=
sprintf(szForm+c,
"Cust-Data: %-
50.50s<BR> %-50.50s<BR> %-
50.50s<BR> %-50.50s<BR>",

```

```

        pPaymentData-
>c_data, pPaymentData->c_data+50,
pPaymentData->c_data+100, pPaymentData-
>c_data+150 );
        else
            strcpy(szForm+c, "Cust-Data:
<BR> <BR> <BR> <BR>");
            strcat(szForm,
" <BR></font></PRE><HR>"
" <INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
" <INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"
" <INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"
" <INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Order-Status..\">"
" <INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Stock-Level..\">"
" <INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Exit..\">"
" </BODY></FORM></HTML>");
        }
}
/* FUNCTION: MakeOrderStatusForm
*
* COMMENTS: The internal client
buffer is created when the terminal id is assigned
and should not
*
be freed except when the client
terminal id is no longer needed.
*/
void MakeOrderStatusForm(int iTermId,
ORDER_STATUS_DATA *pOrderStatusData,
BOOL bInput, char *szForm)
{
    int i, c;
    static char szBR[] = " <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <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=\"\">"
" <INPUT
TYPE=\"hidden\" NAME=\"TERMINID\"
VALUE=\"\">"
" <INPUT
TYPE=\"hidden\" NAME=\"SYNCID\"
VALUE=\"\">"
" <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:
<INPUT NAME=\"CLT*\" SIZE=23><BR>"
"Cust-
Balance:<BR> <BR>"
"Order-
Number: Entry-Date:
Carrier-Number:<BR>"
"Supply-
W Item-Id Qty Amount Delivery-
Date<BR> <BR> <BR> <BR> <BR>"
" <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR></font></PRE>"
" <HR><INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\"><INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Menu\">"
" </BODY></FORM></HTML>" );
    }
    else
    {
        c +=
wsprintf(szForm+c,
"District:
%2.2d<BR>"
"Customer: %4.4d Name: %6-16s
%-2s %6-16s<BR>",
pOrderStatusData->d_id,
pOrderStatusData->c_id,
pOrderStatusData->c_first,
pOrderStatusData->c_middle, pOrderStatusData-
>c_last);
        c +=
sprintf(szForm+c, "Cust-Balance: $%9.2f<BR>
<BR>",
pOrderStatusData->c_balance);
        c +=
wsprintf(szForm+c,

```

```

"Order-
Number: %8.8d Entry-Date: %2.2d-%2.2d-
%4.4d %2.2d:%2.2d:%2.2d Carrier-Number:
%2.2d<BR>"
"Supply-
W Item-Id Qty Amount Delivery-
Date<BR>",
pOrderStatusData->o_id,
pOrderStatusData->o_entry_d.day,
pOrderStatusData-
>o_entry_d.month,
pOrderStatusData->o_entry_d.year,
pOrderStatusData-
>o_entry_d.hour,
pOrderStatusData-
>o_entry_d.minute,
pOrderStatusData-
>o_entry_d.second,
pOrderStatusData->o_carrier_id);
    for(i=0; i<
pOrderStatusData->o_ol_cnt; i++)
    {
        c +=
sprintf(szForm+c, " %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);
    }
    strcpy( 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=\"TERMID\"
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> <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> <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_B
LOCK *pECB, int iTermId, char *szBuffer)
{
    PNEW_ORDER_DATA
pNewOrder;
    pNewOrder =
Term.pClientData[iTermId].pTxn-
>BuffAddr_NewOrder();
}

```

```

ZeroMemory(pNewOrder,
sizeof(NEW_ORDER_DATA));
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_B
LOCK *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
*                               back to
*                               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_BL
OCK *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_B
LOCK *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
                    lpszQueryString
                    client browser http
command string
*
                    PAYMENT_DATA
                    *pPaymentData
                    pointer to payment data structure
*/

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

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

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

```

```

        bCustIdBlank =
FALSE;
        if ( !IsNumeric(szTmp)
)

            throw
new CWEBCLNT_ERR(
ERR_PAYMENT_CUSTOMER_INVALID );
        pPaymentData->c_id
= atoi(szTmp);
    }

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

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

        _strupr( szTmp );

        if ( strlen(szTmp) >
LAST_NAME_LEN )

            throw
new CWEBCLNT_ERR(
ERR_PAYMENT_LAST_NAME_TO_LONG );

        strcpy(pPaymentData-
>c_last, szTmp);
        // pad with spaces so
that the client layer doesn't have to do it
        // before passing
parameters to stored procedure
        iLen =
strlen(pPaymentData->c_last);

        memset(pPaymentData->c_last +
iLen, ' ', LAST_NAME_LEN - iLen);
        pPaymentData-
>c_last[LAST_NAME_LEN] = 0;
        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 =
atoi(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;
    int          iLen;

    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 );

        _strup( szTmp );
        if ( strlen(szTmp) >
LAST_NAME_LEN )
            throw
new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CLT_RANGE );

        strcpy(pOrderStatusData->c_last,
szTmp);
        // pad with spaces so
that the client layer doesn't have to do it
        // before passing
parameters to stored procedure
        iLen =
strlen(pOrderStatusData->c_last);
        memset(pOrderStatusData->c_last
+ iLen, ' ', LAST_NAME_LEN - iLen);
        pOrderStatusData-
>c_last[LAST_NAME_LEN] = 0;
    }
    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_CID_KEY);
        if ( szTmp[0] == 0 )
            throw
new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CID_AND_CLT );
    }
}

/* FUNCTION: BOOL IsNumeric(char *ptr)
*
* PURPOSE: This function determines if a string
is numeric. It fails if any characters other
than
numeric and null terminator are present.
*
* ARGUMENTS:      char        pointer
                  *ptr
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        pointer
                  *ptr
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.rc
//
#define IDD_DIALOG1          101

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

```

#### 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];
    wchar_t szSPPrefix[32];
    //tpcc_odbc.dll stored procedures
prefix
    DWORD dwConnectDelay; // delay
in ms to use in pacing connection open and close
    BOOL bCallNoDuplicatesNewOrder;
    // whether to check for non-
duplicate item ids and call a different New Order
SP
} 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
(dlevine@microsoft.com)
*
* Change history:
* 4.20.000 - first
version
*/

/* FUNCTION: ReadTPCCRegistrySettings

```

```

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

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

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

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

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

```

```

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

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

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

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

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

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

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

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

size = sizeof( pReg->szSPPrefix );
if ( RegQueryValueEx(hKey,
L"SPPrefix", 0, &type, (BYTE *)&pReg-
>szSPPrefix, &size) != ERROR_SUCCESS )

```

```

        pReg->szSPPrefix[0]
= L'\0';

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

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

        RegCloseKey(hKey);

        return FALSE;
}

```

#### common\src\error.h

```

/*      FILE:
      ERROR.H

      *      Microsoft TPC-C Kit Ver. 4.20.000

      *      Copyright Microsoft, 1999
      *      All Rights Reserved

      *

      *      Version 4.10.000 audited by
      Richard Gimarc, Performance Metrics, 3/17/99
      *
      *      PURPOSE:  Header file for error
      exception classes.
      *
      *      Change history:
      *      4.20.000 - updated
      rev number to match kit
      *      4.21.000 - fixed bug:
      ~CBaseErr needed to be declared virtual
      */

#pragma once

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

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

//error message structure used in ErrorText
routines
typedef struct _SERRORMSG
{

```

```

        int                iError;

        char                szMsg[256];
        //error id of message
        //message to sent to
        browser
    } SERRORMSG;

typedef enum _ErrorLevel
{
        ERR_FATAL_LEVEL
            = 1,
        ERR_WARNING_LEVEL
            = 2,
        ERR_INFORMATION_LEVEL
            = 3
    } ErrorLevel;

#define ERR_TYPE_LOGIC
        -1
        //logic error in program; internal
        error
#define ERR_SUCCESS
        0
        //success (a non-error
        error)
#define ERR_BAD_ITEM_ID
        1
        //expected abort
        record in txnRecord
#define ERR_TYPE_DELIVERY_POST
        2
        //expected delivery post failed
#define ERR_TYPE_WEBDLL
        3
        //tpcc web generated
        error
#define ERR_TYPE_SQL
        4
        //sql server generated
        error
#define ERR_TYPE_DBLIB
        5
        //dblib generated
        error
#define ERR_TYPE_ODBC
        6
        //odbc generated
        error
#define ERR_TYPE_SOCKET
        7
        //error on
        communication socket client rte only
#define ERR_TYPE_DEADLOCK
        8
        //dblib and odbc only deadlock
        condition
#define ERR_TYPE_COM
        9
        //error from COM call
#define ERR_TYPE_TUXEDO
        10
        //tuxedo error

```

```

#define ERR_TYPE_OS
        11
        //operating system
        error
#define ERR_TYPE_MEMORY
        12
        //memory allocation
        error
#define ERR_TYPE_TPCC_ODBC
        13
        //error from tpcc odbc txn module
#define ERR_TYPE_TPCC_DBLIB
        14
        //error from tpcc dblib txn module
#define ERR_TYPE_DELISRV
        15
        //delivery server error
#define ERR_TYPE_TXNLOG
        16
        //txn log error
#define ERR_TYPE_BCCONN
        17
        //Benchcraft
        connection class
#define ERR_TYPE_TPCC_CONN
        18
        //Benchcraft connection class
#define ERR_TYPE_ENCINA
        19
        //Encina error
#define ERR_TYPE_COMPONENT
        20
        //error from COM component
#define ERR_TYPE_RTE
        21
        //Benchcraft rte
#define ERR_TYPE_AUTOMATION
        22
        //Benchcraft automation errors
#define ERR_TYPE_DRIVER
        23
        //Driver engine errors
#define ERR_TYPE_RTE_BASE
        24
        //Framework errors
#define ERR_BUF_OVERFLOW
        25
        //Buffer overflow during receive
#define ERR_TYPE_SOAP_HTTP
        26
        //HTTP/SOAP dll generated error
#define ERR_TYPE_OLEDB
        27
        //OLE-DB generated error

```



```

#define ERR_TYPE_TPCC_OLEDB
                28
                //error from tpcc ole-db txn module
// TPC-W error types
#define ERR_TYPE_TPCW_CONN
                50
                //Benchcraft connection class
#define ERR_TYPE_TPCW_HTML
                51
                //error from TpcwHtml dll
#define ERR_TYPE_TPCW_USER
                52
                //error from TPC-W user class
#define ERR_TYPE_TPCW_ENG_BASE
                53
#define ERR_TYPE_TPCW_ENG_OS
                54
#define ERR_TYPE_HTML_RESP
                55
#define ERR_TYPE_TPCW_ODBC
                56
#define ERR_TYPE_SCHANNEL
                57
#define ERR_TYPE_THINK_LIST
                58
//----- end TPC-W -----
#define ERR_TYPE_XML_PROFILE
                59
// TPC-E error types
#define ERR_TYPE_TPCE_CONN
                60
                //TPC-E pipe connection errors
#define ERR_TYPE_TPCE_RTE
                61
                //TPC-E Rte errors
#define ERR_TYPE_TPCE_ENG_BASE
                62
                //Tpce
Driver engine errors
#define ERR_TYPE_TPCE_ENG_OS
                63
                //Tpce
Driver engine system errors
//#define ERR_TYPE_TPCE_MEE_ENG_BASE
                64
                //Tpce
MEE Driver engine errors
//#define ERR_TYPE_TPCE_MEE_ENG_OS
                65
                //Tpce
MEE Driver engine system errors
#define ERR_INS_MEMORY
                "Insufficient Memory
to continue."
#define ERR_UNKNOWN
                "Unknown error."
#define ERR_MSG_BUF_SIZE
                512
#define INV_ERROR_CODE
                -1

```

```

#define ERR_INS_BUF_OVERFLOW
"Insufficient Buffer size to receive HTML pages."

class CBaseErr
{
public:
    enum Action
    {
        eNone = 0
    };

    CBaseErr(LPCTSTR szLoc = NULL)
    {
        m_idMsg
        = GetLastError(); //take
the error code immediately before it is reset by
other functions

        if (szLoc)
        {
            m_szLoc
            = new char[strlen(szLoc)+1/*m_szLoc_size*/];
            strcpy(m_szLoc, szLoc);
        }
        else
            m_szLoc
            = NULL;

        m_szApp
        = new char[m_szApp_size];

        GetModuleFileName(GetModuleHan
dle(NULL), m_szApp, m_szApp_size);
    }

    CBaseErr(int idMsg, LPCTSTR szLoc
= NULL)
    {
        m_idMsg
        = idMsg;

        if (szLoc)
        {
            m_szLoc
            = new char[strlen(szLoc)+1/*m_szLoc_size*/];
            strcpy(m_szLoc, szLoc);
        }
        else
            m_szLoc
            = NULL;

        m_szApp
        = new char[m_szApp_size];

        GetModuleFileName(GetModuleHan
dle(NULL), m_szApp, m_szApp_size);
    }

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

        if (m_szLoc)
            delete []
m_szLoc;
    };
};

```

```

    virtual void Draw(HWND hwnd,
LPCTSTR szStr = NULL)
    {
        int
        j = 0;
        char
        szTmp[512];

        if (szStr)
            j =
            wsprintf(szTmp, "%s\n", szStr);
        if (ErrorNum() !=
INV_ERROR_CODE)
            j +=
            wsprintf(szTmp+j, "Error = %d\n", ErrorNum());
        if (m_szLoc)
            j +=
            wsprintf(szTmp+j, "Location = %s\n",
            GetLocation());

        j +=
            wsprintf(szTmp+j, "%s\n", ErrorText());

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

    char *GetApp(void) { return
m_szApp; }
    char *GetLocation(void) { return
m_szLoc; }
    virtual int ErrorNum() { return
m_idMsg; }

    virtual int ErrorType() = 0; // a
value which distinguishes the kind of error that
occurred

    virtual char *ErrorTypeStr() = 0; //
text representation of the error type

    virtual char *ErrorText() = 0; // a
string (i.e., human readable) representation of
the error

    virtual int ErrorAction() { return
eNone; } // the function call that caused the
error

protected:
    char *m_szApp;
    char *m_szLoc; // code
location where the error occurred
    int
    m_idMsg;

    //short    m_errType;
};

class CSocketErr : public CBaseErr
{
public:
    enum Action
    {
        eNone = 0,
        eSend,
        eSocket,
        eBind,
        eConnect,
        eListen,
        eHost,
        eRecv,
        eGetHostByName,
    };
};

```

```

        eWSACreateEvent,
        eWSASend,

        eWSAGetOverlappedResult,
        eWSARecv,

        eWSAWaitForMultipleEvents,
        eWSAStartup,
        eWSAResetEvent,

        eWSAEnumNetworkEvents,
        eWSAEventSelect,
        eSelect,
        eAccept,
        eNonRetryable
    };

    CSocketErr(Action eAction,
LPCTSTR szLocation = NULL);

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

    Action    m_eAction;
    char      *m_szErrorText;

    int
    ErrType() { return
ERR_TYPE_SOCKET;};
    char*    ErrTypeStr() {
return "SOCKET"; }
    char*    ErrText(void);
    int
    ErrAction() { return
(int)m_eAction; }

};

class CSystemErr : public CBaseErr
{
public:
    enum Action
    {
        eNone = 0,
        eTransactNamedPipe,
        eWaitNamedPipe,

        eSetNamedPipeHandleState,
        eCreateFile,
        eCreateProcess,
        eCallNamedPipe,
        eCreateEvent,
        eCreateThread,
        eVirtualAlloc,
        eReadFile = 10,
        eWriteFile,
        eMapViewOfFile,
        eCreateFileMapping,

        eInitializeSecurityDescriptor,

        eSetSecurityDescriptorDacl,
        eCreateNamedPipe,
        eConnectNamedPipe,
        eWaitForSingleObject,
        eRegOpenKeyEx,
        eRegQueryValueEx =
20,
        eBeginThread,

```

```

        eRegSetValueEx,
        eRegCreateKeyEx,

        eWaitForMultipleObjects,
        eRegisterClassEx,
        eCreateWindow,
        eCreateSemaphore,
        eReleaseSemaphore,
        eFSeek,
        eFRead,
        eFWrite,
        eTmpFile,
        eSetFilePointer,
        eNew,
        eCloseHandle,
        eGetOverlappedResult
    };

    CSystemErr(Action eAction,
LPCTSTR szLocation);

    CSystemErr(int iError, Action
eAction, LPCTSTR szLocation);
    int
    ErrType() { return
ERR_TYPE_OS;};
    char*    ErrTypeStr() {
return "SYSTEM"; }
    char      *ErrText(void);
    int
    ErrAction() { return
(int)m_eAction; }
    void    Draw(HWND hwnd,
LPCTSTR szStr = NULL);

    Action    m_eAction;

private:
    char
    m_szMsg[ERR_MSG_BUF_SIZE];
};

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

    int
    ErrType() {return
ERR_TYPE_MEMORY; }
    char*    ErrTypeStr() {
return "OUT OF MEMORY"; }
    char*    ErrText() {return
ERR_INS_MEMORY; }
};

class CBufferOverflowErr : public CBaseErr
{
public:
    CBufferOverflowErr(int,LPTSTR);

    int
    ErrType() {return
ERR_BUF_OVERFLOW; }
    char*    ErrTypeStr() {
return "BUFFER OVERFLOW"; }
    char*    ErrText() {return
ERR_INS_BUF_OVERFLOW; }
};

// Exception type for XML profiles
class CXMLProfileErr : public CBaseErr
{

```

```

public:
    enum Action
    {
        LoadProfile = 1,
        LoadSchema,
        ValidateProfile,
        SaveProfile,
        LoadFromXML,
        SaveToXML,
        ApplyProcessingInstruction,
        ApplyAttribute,
        ApplyNode
    };

    CXMLProfileErr(Action
eAction, int eCode, LPCTSTR szLocation)
    {
        m_eAction = eAction;
        m_eCode = eCode;
        m_bOverload = true;
    };
    CXMLProfileErr(Action
eAction, int eCode, LPCTSTR szLocation, char *
szMsg)
    {
        m_eAction = eAction;
        m_eCode = eCode;
        strcpy(m_szMsg, szMsg);

        m_bOverload = false;
    };

    virtual int
    ErrType() { return
ERR_TYPE_XML_PROFILE;};
    virtual char
    *ErrTypeStr() { return "XML
PROFILE"; };
    virtual char
    *ErrText();
    virtual int
    ErrCode() { return m_eCode; };
    int

    ErrAction() { return
(int)m_eAction; }
    //virtual void
    Draw(HWND hwnd, LPCTSTR szStr
= NULL)
    {
        // {
        //
        ::MessageBox(hwnd, szStr,
m_szLoc, MB_OK);
        // };
    private:
        char
        m_szMsg[ERR_MSG_BUF_SIZE];
        LPCTSTR m_szLoc;

```

```

        int
        m_eCode;
        bool
        m_bOverload;
        Action
        m_eAction;
};

```

**common\src\trans.h**

```

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

#pragma once

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

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

//error message structure used in ErrorText
//routines
typedef struct _SERRORMSG
{
    int iError;

    //error id of message
    char szMsg[256];
    //message to sent to
    browser
} SERRORMSG;

typedef enum _ErrorLevel
{
    ERR_FATAL_LEVEL
    = 1,
    ERR_WARNING_LEVEL
    = 2,
    ERR_INFORMATION_LEVEL
    = 3
} ErrorLevel;

```

```

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

```

```

#define ERR_TYPE_DELISRV
15
//delivery server error
#define ERR_TYPE_TWLOG
16
//txn log error
#define ERR_TYPE_BCCONN
17
//Benchcraft
connection class
#define ERR_TYPE_TPCC_CONN
18
//Benchcraft connection class
#define ERR_TYPE_ENCINA
19
//Encina error
#define ERR_TYPE_COMPONENT
20
//error from COM component
#define ERR_TYPE_RTE
21
//Benchcraft rte
#define ERR_TYPE_AUTOMATION
22
//Benchcraft automation errors
#define ERR_TYPE_DRIVER
23
//Driver engine errors
#define ERR_TYPE_RTE_BASE
24
//Framework errors
#define ERR_BUF_OVERFLOW
25
//Buffer overflow during receive
#define ERR_TYPE_SOAP_HTTP
26
//HTTP/SOAP dll generated error
#define ERR_TYPE_OLEDB
27
//OLE-DB generated error
#define ERR_TYPE_TPCC_OLEDB
28
//error from tpcc ole-db txn module
// TPC-W error types
#define ERR_TYPE_TPCW_CONN
50
//Benchcraft connection class
#define ERR_TYPE_TPCW_HTML
51
//error from TpcwHtml dll
#define ERR_TYPE_TPCW_USER
52
//error from TPC-W user class
#define ERR_TYPE_TPCW_ENG_BASE
53

```

```

#define ERR_TYPE_TPCW_ENG_OS
    54
#define ERR_TYPE_HTML_RESP
    55
#define ERR_TYPE_TPCW_ODBC
    56
#define ERR_TYPE_SCHANNEL
    57
#define ERR_TYPE_THINK_LIST
    58
//----- end TPC-W -----
#define ERR_TYPE_XML_PROFILE
    59
// TPC-E error types
#define ERR_TYPE_TPCE_CONN
    60
//TPC-E pipe connection errors
#define ERR_TYPE_TPCE_RTE
    61
//TPC-E Rte errors
#define ERR_TYPE_TPCE_ENG_BASE
    62 //Tpce
Driver engine errors
#define ERR_TYPE_TPCE_ENG_OS
    63 //Tpce
Driver engine system errors
//#define ERR_TYPE_TPCE_MEE_ENG_BASE
    64 //Tpce
MEE Driver engine errors
//#define ERR_TYPE_TPCE_MEE_ENG_OS
    65 //Tpce
MEE Driver engine system errors

#define ERR_INS_MEMORY
    "Insufficient Memory
to continue."
#define ERR_UNKNOWN
    "Unknown error."
#define ERR_MSG_BUF_SIZE
    512
#define INV_ERROR_CODE
    -1
#define ERR_INS_BUF_OVERFLOW
    "Insufficient Buffer size to receive HTML pages."

class CBaseErr
{
public:
    enum Action
    {
        eNone = 0
    };

    CBaseErr(LPCTSTR szLoc = NULL)
    {
        m_idMsg
        = GetLastError(); //take
the error code immediatelly before it is reset by
other functions

```

```

        if (szLoc)
            m_szLoc
            = new char[strlen(szLoc)+1/*m_szLoc_size*/];
        strcpy(m_szLoc, szLoc);
    }
    else
        m_szLoc
        = NULL;

        m_szApp
        = new char[m_szApp_size];

        GetModuleFileName(GetModuleHan
dle(NULL), m_szApp, m_szApp_size);
    }

    CBaseErr(int idMsg, LPCTSTR szLoc
= NULL)
    {
        m_idMsg
        = idMsg;

        if (szLoc)
        {
            m_szLoc
            = new char[strlen(szLoc)+1/*m_szLoc_size*/];
            strcpy(m_szLoc, szLoc);
        }
        else
            m_szLoc
            = NULL;

            m_szApp
            = new char[m_szApp_size];

            GetModuleFileName(GetModuleHan
dle(NULL), m_szApp, m_szApp_size);
        }

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

            if (m_szLoc)
                delete []
m_szLoc;
        };

        virtual void Draw(HWND hwnd,
LPCTSTR szStr = NULL)
        {
            int
            j = 0;
            char
            szTmp[512];

            if (szStr)
                j =
                wsprintf(szTmp, "%s\n", szStr);
            if (ErrorNum() !=
INV_ERROR_CODE)
                j +=
                wsprintf(szTmp+j, "Error = %d\n", ErrorNum());
            if (m_szLoc)
                j +=
                wsprintf(szTmp+j, "Location = %s\n",
                GetLocation());

```

```

            j +=
            wsprintf(szTmp+j, "%s\n", ErrorText());
            MessageBox(hwnd,
            szTmp, m_szApp, MB_OK);
        }

        char *GetApp(void) { return
m_szApp; }
        char *GetLocation(void) { return
m_szLoc; }
        virtual int ErrorNum() { return
m_idMsg; }

        virtual int ErrorType() = 0; // a
value which distinguishes the kind of error that
occurred
        virtual char *ErrorTypeStr() = 0; //
text representation of the error type
        virtual char *ErrorText() = 0; // a
string (i.e., human readable) representation of
the error
        virtual int ErrorAction() { return
eNone; } // the function call that caused the
error

protected:
        char *m_szApp;
        char *m_szLoc; // code
location where the error occurred
        int
        m_idMsg;

        //short m_errType;
};

class CSocketErr : public CBaseErr
{
public:
    enum Action
    {
        eNone = 0,
        eSend,
        eSocket,
        eBind,
        eConnect,
        eListen,
        eHost,
        eRecv,
        eGetHostByName,
        eWSACreateEvent,
        eWSASend,
        eWSAGetOverlappedResult,
        eWSARRecv,
        eWSAWaitForMultipleEvents,
        eWSAStartup,
        eWSAResetEvent,
        eWSAEnumNetworkEvents,
        eWSAEventSelect,
        eSelect,
        eAccept,
        eNonRetryable
    };

    CSocketErr(Action eAction,
LPCTSTR szLocation = NULL);

```

```

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

    Action    m_eAction;
    char      *m_szErrorText;

    int
    ErrType() { return
ERR_TYPE_SOCKET;};
    char*     ErrTypeStr() {
return "SOCKET"; }
    char*     ErrText(void);
    int
    ErrAction() { return
(int)m_eAction; }

};

class CSystemErr : public CBaseErr
{
public:
    enum Action
    {
        eNone = 0,
        eTransactNamedPipe,
        eWaitNamedPipe,

        eSetNamedPipeHandleState,
        eCreateFile,
        eCreateProcess,
        eCallNamedPipe,
        eCreateEvent,
        eCreateThread,
        eVirtualAlloc,
        eReadFile = 10,
        eWriteFile,
        eMapViewOfFile,
        eCreateFileMapping,

        eInitializeSecurityDescriptor,

        eSetSecurityDescriptorDacl,
        eCreateNamedPipe,
        eConnectNamedPipe,
        eWaitForSingleObject,
        eRegOpenKeyEx,
        eRegQueryValueEx =
20,

        ebeginthread,
        eRegEnumValue,
        eRegSetValueEx,
        eRegCreateKeyEx,

        eWaitForMultipleObjects,
        eRegisterClassEx,
        eCreateWindow,
        eCreateSemaphore,
        eReleaseSemaphore,
        eFSeek,
        eFRead,
        eFWrite,
        eTmpFile,
        eSetFilePointer,
        eNew,
        eCloseHandle,
        eGetOverlappedResult

    };
};

```

```

    CSystemErr(Action eAction,
LPCTSTR szLocation);
    CSystemErr(int iError, Action
eAction, LPCTSTR szLocation);
    int
    ErrType() { return
ERR_TYPE_OS;};
    char*     ErrTypeStr() {
return "SYSTEM"; }
    char      *ErrText(void);
    int
    ErrAction() { return
(int)m_eAction; }
    void      Draw(HWND hwnd,
LPCTSTR szStr = NULL);

    Action    m_eAction;

private:
    char
m_szMsg[ERR_MSG_BUF_SIZE];
};

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

    int
    ErrType() {return
ERR_TYPE_MEMORY;};
    char*     ErrTypeStr() {
return "OUT OF MEMORY"; }
    char*     ErrText() {return
ERR_INS_MEMORY; }
};

class CBufferOverflowErr : public CBaseErr
{
public:
    CBufferOverflowErr(int,LPTSTR);

    int
    ErrType() {return
ERR_BUF_OVERFLOW;};
    char*     ErrTypeStr() {
return "BUFFER OVERFLOW"; }
    char*     ErrText() {return
ERR_INS_BUF_OVERFLOW; }
};

// Exception type for XML profiles
class CXMLProfileErr : public CBaseErr
{
public:
    enum Action
    {
        LoadProfile = 1,

        LoadSchema,

        ValidateProfile,

        SaveProfile,

        LoadFromXML,

        SaveToXML,

        ApplyProcessingInstruction,

        ApplyAttribute,

```

```

        ApplyNode    };

        CXMLProfileErr(Action
eAction, int eCode, LPCTSTR szLocation)
        {
            m_eAction = eAction;
            m_eCode = eCode;
            m_bOverload = true;
        };
        CXMLProfileErr(Action
eAction, int eCode, LPCTSTR szLocation, char *
szMsg)
        {
            m_eAction = eAction;
            m_eCode = eCode;
            strcpy(m_szMsg, szMsg);

            m_bOverload = false;
        };

        virtual int
        ErrType() { return
ERR_TYPE_XML_PROFILE;};
        virtual char
        *ErrTypeStr() { return "XML
PROFILE"; };
        virtual char
        *ErrText();
        virtual int
        ErrCode() { return m_eCode; };
        int

        ErrAction() { return
(int)m_eAction; }
        //virtual void
        Draw(HWND hwnd, LPCTSTR szStr
= NULL)
        //{
        //
        //::MessageBox(hwnd, szStr,
m_szLoc, MB_OK);
        //};

        private:
            char
m_szMsg[ERR_MSG_BUF_SIZE];
            LPCTSTR m_szLoc;
            int
m_eCode;
            bool
m_bOverload;
            Action
m_eAction;
};

```

Common\src\txn\_base.h

```

/* FILE:
TXN_BASE.H
*
Microsoft TPC-C Kit Ver. 4.20.000

```

```

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

#pragma once

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

class DllDecl CTPCC_BASE
{
public:
    CTPCC_BASE(void)

};

~CTPCC_BASE(void) {};

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

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

```

```

\common\txnlog\include\rtet
ime.h

```

```

/* FILE: rtime.h : header file

```

```

* Copyright 1997 Microsoft Corp., All rights
reserved.
* Source code licensed to Tandem Computers
for Internal
* use only. Redistribution of source or object
files or
* any derivative works is prohibited. By
agreement, this
* notice may not be removed.
*
* Authors: Charles Levine, Philip Durr
*
    Microsoft Corp.
*
*/

//FILE: RTETIME.H

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

```

```

\common\txnlog\include\spin
lock.h

```

```

/* FILE: SPINLOCK.H
*
* Copyright 1997 Microsoft Corp., All rights
reserved.
*
* Source code licensed to Tandem Computers
for Internal
* use only. Redistribution of source or object
files or
* any derivative works is prohibited. By
agreement, this
* notice may not be removed.
*
* Authors: Mike Parkes, Charles Levine, Philip
Durr
*
    Microsoft Corp.
*/

```

```

#ifndef _INC_Spinlock

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

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

class Spinlock
{
    // Private data.
    HANDLE

Semaphore;
    volatile LONG

m_Spinlock;
    volatile LONG

Waiting;

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

    Spinlock( void );

    inline
    BOOL ClaimLock( BOOL Wait = TRUE );
    inline
    void ReleaseLock( void );

    ~Spinlock( void );
//
Disabled operations.

```

```

Spinlock( const Spinlock & Copy );
void
operator=( const Spinlock & Copy );

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

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

*****
*****/

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

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

*****
*****/

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

return TRUE;
}

/*****
*****
*

```

```

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

*****
*****/

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

#define _INC_Spinlock

#endif

\common\txnlog\include\txn1
og.h

/* FILE:
TXNLOG.H
*
Microsoft TPC-C Kit Ver. 4.10.000
*
not yet audited
*
PURPOSE: Header file for txn log
class
*
Copyright Microsoft, 1999
All Rights Reserved
*
*/
#include <stdio.h>
//needed for FILE

#define DRIVER_NAME_LEN
32 //max length of the
driver engine name - must be the same as in
engstut.h!
#define TXN_LOG_INCORRECTLY_SHUT_DOWN
100 //ctrl rec subtype
generated by the txn log when reading an
abruptly shut down log

#pragma once

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

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

typedef struct _TXN_ORDERSTATUS

```

```

{
BYTE
CustByName;
} TXN_ORDERSTATUS;

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

// Common header for all records in
txn log. The TxnType field is
// a switch which identifies the
particular variant.
#define TXN_REC_TYPE_CONTROL
1 //
#define TXN_REC_TYPE_TPCC
2
// replaces
TRANSACTION_TYPE_TPCC
#define
TXN_REC_TYPE_TPCC_DELIV_DEF 3
#define TXN_REC_TYPE_TPCW
4
// replaces
TRANSACTION_TYPE_TPCW

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

typedef struct
_TXN_RECORD_CONTROL
{
// common header;
must exactly match TXN_RECORD_HEADER
of txn
JULIAN_TIME
TxnStartT0; // start
of txn
BYTE
TxnType;
// = TXN_REC_TYPE_CONTROL
BYTE
TxnSubType;
// depends on TxnType
// end of common
header
DWORD Len;
// number of bytes after this field
} TXN_RECORD_CONTROL,
*PTXN_RECORD_CONTROL;

```

```

// TPC-C Txn Record Layout:
//
//TxnStartT0 is a Julian timestamp
corresponding to the moment the
//txn is sent to the SUT, i.e., beginning of
response time. Deltas
//are in milliseconds. Note that if RTDelay >
0, then the txn was
//delayed by this amount. The delay occurs at
the beginning of the
//response time. So if RTDelay > 0, then the
txn was actually sent
//at TxnStartT0 + RTDelay.
//
//Graphically:
//
// time -->
//
// |--- Menu ---|--- Keying ---|--- Response ---
|--- Think ---|
// <- DeltaT1 -> <- DeltaT2 -> <- DeltaT4
-> <- DeltaT3 ->
//
//      ^
//      ^ TxnStartT0
//
//RTDelay is the amount of response time
delay included in DeltaT4.
//RTDelay is recorded per txn because this
value can be changed on
//the fly, and so may vary from txn to txn.
//
//TxnStatus is the txn completion code. It is
used to indicate errors.
//For example, in the New Order txn, 1% of
txns abort. TxnStatus will
//reflect this.

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

    int DeltaT1;
// menu time (ms)
    int DeltaT2;
// keying time (ms)
    int DeltaT3;
// think time (ms)
    int DeltaT4;
// response time (ms)
    int RTDelay;
// response time delay
(ms)

    int
    TxnError; // error
code providing more detail for TxnStatus
    int
    w_id;
// warehouse ID
        BYTE    d_id;
assigned district ID for this thread
}

```

```

        BYTE
        d_id_ThisTxn; //
district ID chosen for this particular
        TxnStatus; //
completion status for txn to indicate errors
        BYTE
        reserved; // for
word alignment
        TXN_DETAILS
        TxnDetails; //

        bool
IsSuccessRecord() { return (TxnStatus ==
ERR_SUCCESS || TxnStatus ==
ERR_BAD_ITEM_ID || TxnStatus ==
ERR_TYPE_DELIVERY_POST); }
        TXN_RECORD_TPCC,
*PTXN_RECORD_TPCC;

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

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

        bool
IsSuccessRecord() { return (TxnStatus ==
ERR_SUCCESS || TxnStatus ==
ERR_BAD_ITEM_ID || TxnStatus ==
ERR_TYPE_DELIVERY_POST); }
        TXN_RECORD_TPCC_DELIV_DEF,
*PTXN_RECORD_TPCC_DELIV_DEF;
}

```

```

//TPC-W records.
//
typedef struct
_TXN_RECORD_TPCW
{
// common header;
must exactly match TXN_RECORD_HEADER
    JULIAN_TIME
of txn    TxnStartT0; // start
        BYTE
        TxnType;
// = TXN_REC_TYPE_TPCW
        BYTE
        TxnSubType;
// depends on TxnType
header    // end of common

    int ThinkTime;
// think time (ms)
    int WIRT;
// response time (ms)
    int
    TxnError; // error
code providing more detail for TxnStatus
        BYTE
        TxnStatus; //
completion status for txn to indicate errors
//This field below
depends on the txn sub type:
// - for Home
interaction: it indicates whether the user was a
new customer (or returning)
// - for Buy Confirm:
it indicates whether
the shipping address was updated
// - for Search
Request: it indicates the search
type (Author, Title, or Subject)
//This statistics needs
to be reported according to 5.5.5.1 clause in the
specs.
//Because this field
occupies 1 byte, the record structure is already
aligned on word boundary.
        union    {
            BYTE
            newCustomer;
            BYTE
            addrUpdated;
            BYTE
            searchType;
        }
        intrDetails;

//This field is mostly
for informational/debugging purposes.
//It indicates what
user performed this web interaction and what
instance (session) of that use it was.
//The first 22 bits
indicate the user #, and the top 10 bits indicate
instance (session) #.
        unsigned __int32
        uiUser;

        bool
IsSuccessRecord() { return (TxnStatus ==
ERR_SUCCESS); }
        TXN_RECORD_TPCW,
*PTXN_RECORD_TPCW;
}

```



```

//
//      Data part of a control
record written when a user is created (or it's new
session) - to record USMD
typedef struct
_TXN_RECORD_TPCW_USER_DATA
{
    unsigned __int32
    uiUser;
    // user number
    JULIAN_TIME
    USMD;

    // USMD for this user
    BYTE

    bRetCust;
    // returning customer?
}
_TXN_RECORD_TPCW_USER_DATA,
*PTXN_RECORD_TPCW_USER_DATA;

//The entire TPCW User control
record structure
typedef struct
_TXN_RECORD_TPCW_USER
{
    // common header;
must exactly match TXN_RECORD_HEADER
    JULIAN_TIME
    TxnStartT0; // start
of txn
    BYTE
    TxnType;
    // = TXN_REC_TYPE_CONTROL
    BYTE
    TxnSubType;
    // depends on TxnType
    // end of common
header
    DWORD Len;

    // number of bytes after this field
    //The fields above
must exactly match TXN_RECORD_CONTROL

    //The fields below
must exactly match
_TXN_RECORD_TPCW_USER_DATA
    unsigned __int32
    uiUser;
    // user number
    JULIAN_TIME
    USMD;

    // USMD for this user
    BYTE

    bRetCust;
    // returning customer?
} TXN_RECORD_TPCW_USER,
*PTXN_RECORD_TPCW_USER;

#define USER_INDEX_NBITS
#define 22 USER_INDEX_MASK
0x003ffff

//lower 22 bits mask for user field
in TPCW record
#define USER_SESSION_MASK
0xffc00000 //upper
10 bits mask for user field in TPCW record

```

```

#define USER_CREATE_REC
254
//subtype for the control record
written when a user is created
#define TXN_LOG_VERSION
2
#define TXN_DATA_START
4096 // offset
in log file where log records start
#define TXN_LOG_EYE_CATCHER "BC"
// signature bytes at the start of log
file

////////////////////////////////////
////////////////////////////////////
// The transaction log has a header
as the first 4K block.
//
typedef struct _TXN_LOG_HEADER
{
    char
    EyeCatcher[2];
be "BC" // signature bytes; should always
    int
    LogVersion; // set to
TXN_LOG_VERSION
    JULIAN_TIME
    BeginTxnTS;
// timestamp of first (lowest) txn
start
    JULIAN_TIME
    EndTxnTS;
// timestamp of last (highest) txn
completion time
    int
    iRecCount; //
number of records in log file
    BOOL
    bLogSorted;
    int
    iFileSize;
// file size in bytes
// driver engine that
created this log file
    char
    szDriverEngineName[DRIVER_NAM
E_LEN];
// the record map
provides a fast way to get close to a particular
timestamp in a sorted log file.
//
// struct
// {
//
    JULIAN_TIME
    TS;
// timestamp of record
    int
    iPos; // byte
position in file
//
}
RecMap[RecMapSize];
// #define RecMapSize
200
} TXN_LOG_HEADER,
*PTXN_LOG_HEADER;

```

```

/* Header of the sorted pointers
blocks in Temp file (in merging). */
typedef struct BLOCK_HEADER {
    long
    BlockPos;
    __int64 CurPos;
    DWORD
    BytesRead;
    int
    nRecords;
    BYTE *offset;
// offset of pointers to records in
the log file */
} BLOCK_HEADER,
*PBLOCK_HEADER;

#define READ_BUFFER_SIZE
64*1024
// #define WRITE_BUFFER_SIZE
8*1024
#define WRITE_BUFFER_SIZE
128*1024

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

// flags passed in to the constructor
#define TXN_LOG_WRITE
0x01
#define TXN_LOG_READ
0x02
#define TXN_LOG_SORTED
0x04
#define TXN_LOG_CRASHOPEN 0x08
// if set, invalid
headers will be tolerated; used for recovery

#define TXN_LOG_OS_ERROR 1
#define TXN_LOG_NOT_SORTED 2

#define SKIP_CTRL_RECS 1

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

```

```

        LARGE_INTEGER
        IFilePointer;
        //position in file.
        int
        iNextRec;
        //when
        reading, ordinal value of next record

        // A "save point" is
        remembered each time GetNextRecord is called
        with a start time specified.
        // The next time it is
        called, if start time is after the save point, we
        start scanning from the
        // save point. This is
        particularly useful in FindBestInterval, where the
        log is scanned repeatedly.
        JULIAN_TIME
        SavePtTime;
        //
        ISavePtFilePointer;
        LARGE_INTEGER
        ISavePtFilePointer;
        int

        ISavePtNextRec;

        JULIAN_TIME
        lastTS;
        //when
        writing sorted output, used to verify records are
        sorted
        BOOL
        bWrite;
        //writing
        log file
        BOOL
        bCrashOpen;
        // tolerate bad
        headers and consistency checks
        BOOL
        bLogSorted;
        // is log file sorted?
        applies to both input and output
        JULIAN_TIME
        BeginTxnTS;
        // timestamp of first
        (lowest) txn start
        JULIAN_TIME
        EndTxnTS;
        // timestamp of last
        (highest) txn completion time
        int

        iRecCount;
        // number of records in log file
        // To write a
        checkpoint information into the header, need to
        know the EndTxnTS for the
        // last record written
        to the disk. It is not necessarily the last record in
        the
        // last written buffer,
        as the last record may be only partially in the
        buffer.
        // So remember the
        timestamps for 2 last records that begin in the
        buffer - one of
        // them will be the last
        complete record written to disk.

```

```

        JULIAN_TIME
        PrevEndTxnTS;
        // timestamp of the
        union {
        previous to last record

        TXN_LOG_HEADER
        HeaderForCheckpoint;// header
        written on every checkpoint
        char
        szHeaderBuffer[512]; // 512
        bytes is the minimum we can write to the disk
        } HeaderBuffer;
        //need the union because can't
        write sizeof(TXN_LOG_HEADER) - too few bytes

        // Control record
        returned from GetNextRecord if the file
        // currently opened
        for read was not properly shut down
        struct
        {
        TXN_RECORD_CONTROL
        RecHeader;
        char
        szDriverName[DRIVER_NAME_LEN]
        ;
        }
        IncorrectShutDownRec;

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

        PTXN_RECORD_HEADER
        *TxnArray;
        //transaction record pointer array
        for sort
        DWORD
        dwError;
        DWORD
        dwCheckpointError;
        //error in checkpoint thread
        HANDLE
        hTxnFile;
        //handle to log file
        HANDLE
        hMapFile;
        //map file used when
        sorting the log
        HANDLE
        hIoComplete;
        //event to signify that there are no
        pending IOs
        HANDLE
        hLogFileIo;
        //event to signal the
        IO thread to write the inactive buffer
        HANDLE
        hStopCheckpointThread; //event
        to signal the checkpoint thread to exit
        Spinlock Spin;
        //spin lock to protect
        the txn log file buffers

```

```

        Spinlock
        WriteSpin;
        //spin lock to protect
        the WriteFile operation between IO and
        Checkpoint threads
        FILE
        *tmpFile;
        //temp
        file for merging sorted pieces
        PBLOCK_HEADER
        tmpHeaders;
        //sorted pointers block header
        BYTE
        **recPointers;
        //record
        pointer buffers for each sorted block
        PTXN_RECORD_HEADER
        *recBuffers; //record buffers for
        each sorted block
        int
        *PointersRead;
        // # of
        pointers processed in each block
        BOOL
        *BlockAvailable;
        //whether to check a
        particular block for jmin
        int
        nBlocks;
        int
        jmin;
        //index (block-wise) of the lowest
        timestamp record
        int
        iAvgRecordLen;
        //average record length
        int
        iSortedReturnedCount;
        //keeps track of the # of sorted
        records returned through GetSortedRecord()
        BOOL
        bIncorrectShutDown;
        // indicates whether
        the log opened for read was not correctly shut
        down
        int Write(BYTE *ptr,
        DWORD Size);
        static void
        LogFileIO(CTxnLog *);
        void LoadBuffers(int
        j);
        //used in
        sort/merge to load record buffers
        static void
        CheckpointThread(CTxnLog *);
        //
        checkpointing thread
        public:
        CTxnLog(LPCTSTR
        szFileName, DWORD dwOpts, char *szDriver =
        NULL);
        ~CTxnLog(void);

```

```

        int
WriteToLog(PTXN_RECORD_TPCC pTxnRcrd);
        int
WriteToLog(PTXN_RECORD_TPCC_DELIV_DEF
pTxnRcrd);
        int
WriteToLog(PTXN_RECORD_CONTROL pCtrlRec);
        int
WriteToLog(PTXN_RECORD_HEADER pCtrlRec);
        int
WriteToLog(PTXN_RECORD_TPCW pTxnRcrd);
//support for TPC-W

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

        void
CloseTransactionLogFile(void);

        PTXN_RECORD_HEADER
GetNextRecord(BOOL bSkipCtrlRecs = FALSE);

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

        int Sort(void);

        PTXN_RECORD_HEADER
GetSortedRecord();

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

class CTXNLOG_ERR : public CBaseErr
{
public:
        enum
CTXNLOG_ERRS
        {
                ERR_BAD_FILE_FORMAT,
                // "File format is invalid."

                ERR_UNKNOWN_LOG_VERSION,
                // "Log file version is unknown."

                ERR_BROKEN_LOG_FILE,
                // "Log file is broken."

                ERR_LOG_NOT_SORTED,
                // "Log file is not
sorted"

                ERR_INVALID_TIME_SEQ,
                // "Internal Error: Record Time
Sequence invalid."
        };

        CTXNLOG_ERR(int
iErr) : CBaseErr(iErr) {};

        int ErrorType()
{return ERR_TYPE_TXNLOG;};

```

```

        char *ErrorTypeStr() {
return "TXN LOG"; }
        char *ErrorText()
{
        static
char *szMsgs[] = {
                "File format is invalid.",
                "Log file version is unknown.",
                "Log file is broken.",
                "Log file is not sorted",
                "Internal Error: Record Time
Sequence invalid.",
                ""
        };
        for(int i
= 0; szMsgs[i][0]; i++)
        {
                if ( m_idMsg == i )
                        break;
        }
        return(szMsgs[i][0] ? szMsgs[i] :
ERR_UNKNOWN);
};

```

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

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

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

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

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

#define DEFCLPACKSIZE
4096

// version string; must match return value from
tpcc_version stored proc
const char sVersion[] = "4.20.000";

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

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

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

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

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

int err_handler(DBPROCESS *dbproc, int severity,
int dberr, int oserr, LPCSTR dberrstr, LPCSTR
oserrstr)

```

```

{
    CTPCC_DBLIB
        *pConn;

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

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

/* FUNCTION: int msg_handler(DBPROCESS
*dbproc, DBINT msgno, int msgstate, int
severity, char *msgtext)
*
* PURPOSE: This function handles DB-Library
SQL Server error messages
*
* ARGUMENTS:      DBPROCESS
                  *dbproc
                  DBPROCESS id pointer
*
                  DBINT
                  msgno
                  message number
*
                  int
                  msgstate
                  message state
*
                  int
                  severity
                  message severity
*
                  char
                  *msgtext          printable
                  message description
*
* RETURNS:        int

                  INT_CONTINUE      continue
if error is SQLTIME else INT_CANCEL action
*

                  INT_CANCEL
                  cancel operation
*
* COMMENTS:      This function also sets
the dead lock dbproc variable if necessary.
*
*/

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

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

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

```

```

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

        return 0;
    }

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

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

    return;
}

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
*
*/

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

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

```

```

        { 0,
        ""
        };
    }

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

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

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

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

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

    m_MaxRetries = 10;
    // how many retries on deadlock

```

```

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

// 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::eDbProcH
andler);

if (dbprocmsgshandle(login,
msg_handler) == NULL)

    ThrowError(CDBLIBERR::eDbProcH
andler);

    DBSETUSER(login, szUser);
    DBSETLPWD(login, szPassword);
    DBSETLHOST(login, szHost);
    DBSETLPACKET(login, (unsigned
short)DEFCLPCKSIZE);
    DBSETLVERSION(login, DBVER60);
    // use dblink 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

```

```

== FAIL)
if (dbuse(m_dbproc, szDatabase)
ThrowError(CDBLIBERR::eDbUse);

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

if (dbsqlxec(m_dbproc) == FAIL)

    ThrowError(CDBLIBERR::eDbSqlExe
c);

    DiscardNextResults(2);

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

if (dbrcpexec(m_dbproc) == FAIL)

    ThrowError(CDBLIBERR::eDbRpcEx
ec);

if (dbresults(m_dbproc) !=
SUCCEEDED)

    ThrowError(CDBLIBERR::eDbResult
s);

if (dbnextrow(m_dbproc) !=
REG_ROW)

    ThrowError(CDBLIBERR::eDbNextR
ow);

    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 DLib 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::eDbNextR
                    ow);
                else
                    break;
            }
            iRowsRead++;
        }

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

    // 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
        iRowsRead = 0;
        RETCODE rc;

        while (TRUE)
        {
            rc =
            dbresults(m_dbproc);
            if (rc ==
            NO_MORE_RESULTS)
                break;
            if (rc == FAIL)
            {
                if
                (iExpectedCount >= 0)
                    ThrowError(CDBLIBERR::eDbResult
                    s);
                else
                    break;
            }
            DiscardNextRows(-1);
            iResultsRead++;
        }

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

    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::eDbRpcEx
                    ec);

                if
                (dbresults(m_dbproc) != SUCCEED)

```

```

                ThrowError(CDBLIBERR::eDbResult
                s);
                if
                (dbnextrow(m_dbproc) != REG_ROW)
                    ThrowError(CDBLIBERR::eDbNextR
                    ow);
                if
                (pData=dbdata(m_dbproc, 1))
                    m_txn.StockLevel.low_stock =
                    *((long *) pData);

                DiscardNextRows(0);
                DiscardNextResults(0);

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

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

            int
            iTryCount = 0;
            const BYTE *pData;

            ResetError();

```

```

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

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

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

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

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

        // check
whether any order lines are for a remote
warehouse

        m_txn.NewOrder.o_all_local = 1;
        for (i =
0; i < m_txn.NewOrder.o_ol_cnt; i++)
        {
            if
(m_txn.NewOrder.OL[i].ol_supply_w_id !=
m_txn.NewOrder.w_id)
            {
                m_txn.NewOrder.o_all_local = 0;
// at least one remote warehouse

                break;
            }
        }

        dbrpcparam(m_dbproc, NULL, 0,
SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.o_all_local);

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

            dbrpcparam(m_dbproc, NULL, 0,
SQLINT4, -1, -1, (BYTE *)
&m_txn.NewOrder.OL[i].ol_supply_w_id);

            dbrpcparam(m_dbproc, NULL, 0,
SQLINT2, -1, -1, (BYTE *)
&m_txn.NewOrder.OL[i].ol_quantity);
        }

        if
(dbrpcexec(m_dbproc) == FAIL)

```

```

);
        ThrowError(CDBLIBERR::eDbRpcEx
ec);

        // 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::eDbResult
s);

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

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

            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::eDbResult
s);

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

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

    if
(pData=dbdata(m_dbproc, 1))

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

    if
(pData=dbdata(m_dbproc, 2))

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

    if
(pData=dbdata(m_dbproc, 3))

        m_txn.NewOrder.o_id = (*(DBINT
*) pData);

    if
(pData=dbdata(m_dbproc, 4))

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

    if
(pData=dbdata(m_dbproc, 5))

```

```

        dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,5), SQLFLT8, (BYTE
*)&m_bxn.NewOrder.c_discount, 8);
        if
(pData=dbdata(m_dbproc, 6))
            UtilStrCpy(m_bxn.NewOrder.c_credi
t, pData, dbdatlen(m_dbproc, 6));
        if
(pData=dbdata(m_dbproc, 7))
        {
            datetime = *((DBDATETIME *)
pData);
            dbdatecrack(m_dbproc, &daterec,
&datetime);
            m_bxn.NewOrder.o_entry_d.year
= daterec.year;
            m_bxn.NewOrder.o_entry_d.month
= daterec.month;
            m_bxn.NewOrder.o_entry_d.day
= daterec.day;
            m_bxn.NewOrder.o_entry_d.hour
= daterec.hour;
            m_bxn.NewOrder.o_entry_d.minute
= daterec.minute;
            m_bxn.NewOrder.o_entry_d.second
= daterec.second;
        }
        if
(pData=dbdata(m_dbproc, 8))
            commit_flag = *((DBTINYINT *)
pData);
            DiscardNextRows(0);
            DiscardNextResults(0);
            if
(commit_flag == 1)
            {
                m_bxn.NewOrder.total_amount *=
((1 + m_bxn.NewOrder.w_tax +
m_bxn.NewOrder.d_tax) * (1 -
m_bxn.NewOrder.c_discount));
                m_bxn.NewOrder.exec_status_code
= eOK;
            }
            else
                m_bxn.NewOrder.exec_status_code
= eInvalidItem;
            return;
        }
        catch (CSQLERR *e)
        {
            if ((e-
>m_msgno == 1205 ||

```

```

        (e->m_msgno ==
iErrOleDbProvider &&
strstr(e->m_msgtext,
sErrTimeoutExpired) != NULL)) &&
        {
            (++iTryCount <= iMaxRetries)
            {
                // hit deadlock; backoff for
increasingly longer period
                delete e;
                Sleep(10 * iTryCount);
            }
            else
                throw;
        }
        // while (TRUE)
        {
            // if (iTryCount)
            // throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RE
TRIED_TRANS, iTryCount);
        }
        void CTPCC_DBLIB::Payment()
        {
            DBDATETIME datetime;
            DBDATERECEC daterec;
            int
iTryCount = 0;
            const BYTE *pData;
            ResetError();
            while (TRUE)
            {
                try
                {
                    dbrpcinit(m_dbproc,
"tpcc_payment", 0);
                    dbrpcparam(m_dbproc, NULL, 0,
SQLINT4, -1, -1, (BYTE *)
&m_bxn.Payment.w_id);
                    dbrpcparam(m_dbproc, NULL, 0,
SQLINT4, -1, -1, (BYTE *)
&m_bxn.Payment.c_w_id);
                    dbrpcparam(m_dbproc, NULL, 0,
SQLFLT8, -1, -1, (BYTE *)
&m_bxn.Payment.h_amount);
                    dbrpcparam(m_dbproc, NULL, 0,
SQLINT1, -1, -1, (BYTE *)
&m_bxn.Payment.d_id);
                    dbrpcparam(m_dbproc, NULL, 0,
SQLINT1, -1, -1, (BYTE *)
&m_bxn.Payment.c_d_id);
                    dbrpcparam(m_dbproc, NULL, 0,
SQLINT4, -1, -1, (BYTE *)
&m_bxn.Payment.c_id);
                }
                // if
customer id is zero, then payment is by name
            }
        }
    }
}

```

```

        if
(m_bxn.Payment.c_id == 0)
            dbrpcparam(m_dbproc, NULL, 0,
SQLCHAR, -1, strlen(m_bxn.Payment.c_last),
(unsigned char *)m_bxn.Payment.c_last);
        if
(dbrpcexec(m_dbproc) == FAIL)
            ThrowError(CDBLIBERR::eDbRpcEx
ec);
        if
(dbresults(m_dbproc) != SUCCEEDED)
            ThrowError(CDBLIBERR::eDbResult
s);
        if
(dbnextrrow(m_dbproc) != REG_ROW)
            ThrowError(CDBLIBERR::eDbNextR
ow);
        if
(dbnumcols(m_dbproc) != 27)
            ThrowError(CDBLIBERR::eWrongNu
mCols);
        if
(pData=dbdata(m_dbproc, 1))
            m_bxn.Payment.c_id = *((DBINT *)
pData);
        if
(pData=dbdata(m_dbproc, 2))
            UtilStrCpy(m_bxn.Payment.c_last,
pData, dbdatlen(m_dbproc, 2));
        if
(pData=dbdata(m_dbproc, 3))
        {
            datetime = *((DBDATETIME *)
pData);
            dbdatecrack(m_dbproc, &daterec,
&datetime);
            m_bxn.Payment.h_date.year =
daterec.year;
            m_bxn.Payment.h_date.month =
daterec.month;
            m_bxn.Payment.h_date.day =
daterec.day;
            m_bxn.Payment.h_date.hour =
daterec.hour;
            m_bxn.Payment.h_date.minute =
daterec.minute;
            m_bxn.Payment.h_date.second =
daterec.second;
        }
        if
(pData=dbdata(m_dbproc, 4))
            UtilStrCpy(m_bxn.Payment.w_street
_1, pData, dbdatlen(m_dbproc, 4));
    }
}

```



```

        if
(pData=dbdata(m_dbproc, 5))
        UtilStrCpy(m_txn.Payment.w_street
_2, pData, dbdatlen(m_dbproc, 5));
        if
(pData=dbdata(m_dbproc, 6))
        UtilStrCpy(m_txn.Payment.w_city,
pData, dbdatlen(m_dbproc, 6));
        if
(pData=dbdata(m_dbproc, 7))
        UtilStrCpy(m_txn.Payment.w_state,
pData, dbdatlen(m_dbproc, 7));
        if
(pData=dbdata(m_dbproc, 8))
        UtilStrCpy(m_txn.Payment.w_zip,
pData, dbdatlen(m_dbproc, 8));
        if
(pData=dbdata(m_dbproc, 9))
        UtilStrCpy(m_txn.Payment.d_street
_1, pData, dbdatlen(m_dbproc, 9));
        if
(pData=dbdata(m_dbproc, 10))
        UtilStrCpy(m_txn.Payment.d_street
_2, pData, dbdatlen(m_dbproc, 10));
        if
(pData=dbdata(m_dbproc, 11))
        UtilStrCpy(m_txn.Payment.d_city,
pData, dbdatlen(m_dbproc, 11));
        if
(pData=dbdata(m_dbproc, 12))
        UtilStrCpy(m_txn.Payment.d_state,
pData, dbdatlen(m_dbproc, 12));
        if
(pData=dbdata(m_dbproc, 13))
        UtilStrCpy(m_txn.Payment.d_zip,
pData, dbdatlen(m_dbproc, 13));
        if
(pData=dbdata(m_dbproc, 14))
        UtilStrCpy(m_txn.Payment.c_first,
pData, dbdatlen(m_dbproc, 14));
        if
(pData=dbdata(m_dbproc, 15))
        UtilStrCpy(m_txn.Payment.c_middle
, pData, dbdatlen(m_dbproc, 15));
        if
(pData=dbdata(m_dbproc, 16))
        UtilStrCpy(m_txn.Payment.c_street
_1, pData, dbdatlen(m_dbproc, 16));
        if
(pData=dbdata(m_dbproc, 17))
        UtilStrCpy(m_txn.Payment.c_street
_2, pData, dbdatlen(m_dbproc, 17));
        if
(pData=dbdata(m_dbproc, 18))
        UtilStrCpy(m_txn.Payment.c_city,
pData, dbdatlen(m_dbproc, 18));
        if
(pData=dbdata(m_dbproc, 19))
        UtilStrCpy(m_txn.Payment.c_state,
pData, dbdatlen(m_dbproc, 19));

```

```

        if
(pData=dbdata(m_dbproc, 20))
        UtilStrCpy(m_txn.Payment.c_zip,
pData, dbdatlen(m_dbproc, 20));
        if
(pData=dbdata(m_dbproc, 21))
        UtilStrCpy(m_txn.Payment.c_phone,
pData, dbdatlen(m_dbproc, 21));
        if
(pData=dbdata(m_dbproc, 22))
        {
            datetime = *((DBDATETIME *)
pData);
            dbdatecrack(m_dbproc, &daterec,
&datetime);
            m_txn.Payment.c_since.year =
daterec.year;
            m_txn.Payment.c_since.month =
daterec.month;
            m_txn.Payment.c_since.day =
daterec.day;
            m_txn.Payment.c_since.hour =
daterec.hour;
            m_txn.Payment.c_since.minute =
daterec.minute;
            m_txn.Payment.c_since.second =
daterec.second;
        }
        if(pData=dbdata(m_dbproc, 23))
        UtilStrCpy(m_txn.Payment.c_credit,
pData, dbdatlen(m_dbproc, 23));
        if(pData=dbdata(m_dbproc, 24))
            dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,24), SQLFLT8, (BYTE
*)&m_txn.Payment.c_credit_lim, 8);
        if(pData=dbdata(m_dbproc, 25))
            dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,25), SQLFLT8, (BYTE
*)&m_txn.Payment.c_discount, 8);
        if(pData=dbdata(m_dbproc, 26))
            dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,26), SQLFLT8, (BYTE
*)&m_txn.Payment.c_balance, 8);
        if(pData=dbdata(m_dbproc, 27))
            UtilStrCpy(m_txn.Payment.c_data,
pData, dbdatlen(m_dbproc, 27));
        DiscardNextRows(0);
        DiscardNextResults(0);

```

```

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

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

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

    ResetError();

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

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

```

```

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

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

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

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

if
(dbrpcexec(m_dbproc) == FAIL)
        ThrowError(CDBLIBERR::eDbRpcExc
ec);

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

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

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

if(pData=dbdata(m_dbproc, 1))

        m_txn.OrderStatus.OL[i].ol_supply_
w_id = (*(DBSMALLINT *) pData);

```

```

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

        if(pData=dbdata(m_dbproc, 3))

                m_txn.OrderStatus.OL[i].ol_quantit
y = (*(DBSMALLINT *) pData);

        if(pData=dbdata(m_dbproc, 4))

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

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

        if(pData=dbdata(m_dbproc, 5))
        {
                datetime =
(*(DBDATETIME *) pData);

                dbdatecrack(m_dbproc, &daterec,
&datetime);

                m_txn.OrderStatus.OL[i].ol_delivery
_d.year = daterec.year;

                m_txn.OrderStatus.OL[i].ol_delivery
_d.month = daterec.month;

                m_txn.OrderStatus.OL[i].ol_delivery
_d.day = daterec.day;

                m_txn.OrderStatus.OL[i].ol_delivery
_d.hour = daterec.hour;

                m_txn.OrderStatus.OL[i].ol_delivery
_d.minute = daterec.minute;

                m_txn.OrderStatus.OL[i].ol_delivery
_d.second = daterec.second;
        }
        i++;
}

        m_txn.OrderStatus.o_ol_cnt = i;

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

if
(dbnextrow(m_dbproc) != REG_ROW)

```

```

        ThrowError(CDBLIBERR::eDbNextR
ow);
if
(dbnumcols(m_dbproc) != 8)
        ThrowError(CDBLIBERR::eWrongNu
mCols);

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_firs
t, pData, dbdatlen(m_dbproc,3));

if(pData=dbdata(m_dbproc, 4))
        UtilStrCpy(m_txn.OrderStatus.c_mi
ddle, 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.mont
h = 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.minut
e = daterec.minute;
        m_txn.OrderStatus.o_entry_d.secon
d = 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_bxn.OrderStatus.o_id =
        (*(DBINT *) pData);

        DiscardNextRows(0);

        DiscardNextResults(0);

        if
        (m_bxn.OrderStatus.o_ol_cnt == 0)

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

            throw new CTPCC_DBLIB_ERR(
            CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
        else

            m_bxn.OrderStatus.exec_status_cod
e = eOK;

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

            (e->m_msgno ==
iErrOleDbProvider &&

            strstr(e->m_msgtext,
sErrTimeoutExpired) != NULL)) &&

            (++iTryCount <= iMaxRetries))
            {
                // hit deadlock; backoff for
                increasingly longer period

                delete e;

                Sleep(10 * iTryCount);
            }
            else

                throw;
        }
        // while (TRUE)

        // if (iTryCount)
        // throw new
        CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RE
        TRIED_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_bxn.Delivery.w_id);

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

                if
                (dbrpcexec(m_dbproc) == FAIL)

                    ThrowError(CDBLIBERR::eDbRpcEx
                    ec);

                if
                (dbresults(m_dbproc) != SUCCEED)

                    ThrowError(CDBLIBERR::eDbResult
                    s);

                if
                (dbnextrow(m_dbproc) != REG_ROW)

                    ThrowError(CDBLIBERR::eDbNextR
                    ow);

                if
                (dbnumcols(m_dbproc) != 10)

                    ThrowError(CDBLIBERR::eWrongNu
                    mCols);

                for (i=0;
                i<10; i++)

                    if (pData = dbdata(m_dbproc, i+1))

                        m_bxn.Delivery.o_id[i]
                        = (*(DBINT *)pData);

                DiscardNextRows(0);

                DiscardNextResults(0);

                m_bxn.Delivery.exec_status_code =
                eOK;

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

                (e->m_msgno ==
                iErrOleDbProvider &&

                strstr(e->m_msgtext,
                sErrTimeoutExpired) != NULL)) &&

                (++iTryCount <= iMaxRetries))
            }
        }
    }

```

```

        {
            // hit deadlock; backoff for
            increasingly longer period

            delete e;

            Sleep(10 * iTryCount);
        }
        else

            throw;
    }
    // while (TRUE)

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

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

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

Db dblib tpcc_dblib.h

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

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

```

```

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

#define DllDecl __declspec( dllimport )
#endif

class CSQLERR : public CBaseErr
{
public:
    CSQLERR(void)
    {
        m_msgno = 0;
        m_msgstate = 0;
        m_severity = 0;
        m_msgtext = NULL;
    };
    ~CSQLERR()
    {
        delete []
        m_msgtext;
    };
    int
    m_msgno;
    int
    m_msgstate;
    int
    m_severity;
    char *m_msgtext;

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

class CDBLIBERR : public CBaseErr
{
public:
    enum ACTION
    {
        eNone,
        eUnknown,
        eLogin,
        // error from dblogin
        eDbOpen,
        // error from dbopen
        eDbUse,
        // error from dbuse
        eDbSqlExec,
        // error from dbsqlexec
        eDbSet,
        // error from one of the dbset*
        routines
    };

```

```

        eDbNextRow,
        // error from dbnextrow
        eWrongRowCount,
        // more or less rows returned than
        expected
        eWrongNumCols,
        // more or less columns returned
        than expected
        eDbResults,
        // error from dbresults
        eDbRpcExec,
        // error from dbrpcexec
        eDbSetMaxProcs,
        // error from dbsetmaxprocs
        eDbProcHandler
        // error from either dbprocerrhandle
        or dbprocmsghandle
    };
    CDBLIBERR(ACTION
    eAction, int severity = 0, int dberror = 0, int
    oserr = 0)
    {
        m_eAction = eAction;
        m_severity = severity;
        m_dberror = dberror;
        m_oserr
        = oserr;
        m_dberrstr = NULL;
        m_oserrstr = NULL;
    };
    ~CDBLIBERR()
    {
        delete []
        m_dberrstr;
        delete []
        m_oserrstr;
    };
    ACTION
    m_eAction;
    int
    m_severity;
    int
    m_dberror;
    int
    m_oserr;
    char *m_dberrstr;
    char *m_oserrstr;

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

```

```

        int
        ErrorAction() { return
        (int)m_eAction; }
    };
    class CTPCC_DBLIB_ERR : public CBaseErr
    {
    public:
        enum
        CTPCC_DBLIB_ERRS
        {
            ERR_WRONG_SP_VERSION = 1,
            // "Wrong version of stored procs
            on database server"
            ERR_INVALID_CUST,
            // "Invalid Customer
            id,name."
            ERR_NO_SUCH_ORDER,
            // "No orders found
            for customer."
            ERR_RETRIED_TRANS,
            // "Retries before
            transaction succeeded."
        };
        CTPCC_DBLIB_ERR(
        int iErr ) { m_erno = iErr; m_iTryCount = 0; };
        CTPCC_DBLIB_ERR(
        int iErr, int iTryCount ) { m_erno = iErr;
        m_iTryCount = iTryCount; };
        int
        m_erno;
        int
        m_iTryCount;

        int
        ErrorType() {return
        ERR_TYPE_TPCC_DBLIB;};
        char*
        ErrorTypeStr() { return "TPCC
        DBLIB"; }
        int
        ErrorNum() {return m_erno;};
        char*
        ErrorText();
    };
    class DllDecl CTPCC_DBLIB : public CTPCC_BASE
    {
    private:
        // declare variables
        and private functions here...
        PDBPROCESS
        m_dbproc;
        CDBLIBERR
        *m_DbLibErr;
        // not allocated until needed
        (maybe never)
        CSQLERR
        *m_SqlErr;
        // not allocated until needed
        (maybe never)
        int
        m_MaxRetries;
        // retry count on
        deadlock
    };

```

```

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

        union
        {

NEW_ORDER_DATA
NewOrder;

PAYMENT_DATA
Payment;

DELIVERY_DATA
Delivery;

STOCK_LEVEL_DATA
StockLevel;

ORDER_STATUS_DATA
OrderStatus;
        }
        m_txn;

public:

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

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

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

        // these are public
because they must be called from the dblib
err_handler and msg_hangler
        // outside of the class

```

```

        void SetDbLibError(int
severity, int dberr, int oserr, LPCSTR dberrstr,
LPCSTR oserrstr);
        void SetSqlError( int
msgno, int msgstate, int severity, LPCSTR
msgtext );

};

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

```

```

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

```

```

db_odbc.dll tpc_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>

#ifdef COMPILE_FOR_SNAC // define
that to compile for SQL Native Client; comment
out to use MDAC

#ifndef COMPILE_FOR_SNAC
#include <odbcss.h>
#else
// Compile for SNAC
#include <sqlncli.h>
#endif
#endif

```

```

#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[] = "4.20.000";

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

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

static SQLHENV henv = SQL_NULL_HENV;

// ODBC environment handle

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

                DisableThreadLibraryCalls(hModule)
;
                if (
SQLAllocHandleStd(SQL_HANDLE_ENV,
SQL_NULL_HANDLE, &henv) != SQL_SUCCESS )
                return FALSE;
                break;
                case
DLL_PROCESS_DETACH:
                if (henv
!= NULL)
                SQLFreeEnv(henv);
                break;
                default:
                /*
nothing */;
        }
        return TRUE;
}

/* FUNCTION: CTPCC_ODBC_ERR::ErrorText
*
*/

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

        static SERRORMSG errorMsgs[] =
        {
                {
ERR_WRONG_SP_VERSION,
"Wrong version of stored procs on
database server"
},

```

```

        {
ERR_INVALID_CUST,
    "Invalid Customer id,name."
        },
        {
ERR_NO_SUCH_ORDER,
orders found for customer."
        },
        {
ERR_RETRIED_TRANS,
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
    LPCWSTR szSPPrefix, // prefix
to append to the stored procedure names
    BOOL bCallNoDuplicatesNewOrder )
// whether to check for non-duplicate items in
NewOrder and call a new SP
{
    return new CTPCC_ODBC( szServer,
szUser, szPassword, szHost, szDatabase,
szSPPrefix, bCallNoDuplicatesNewOrder );
}

CTPCC_ODBC::CTPCC_ODBC (
    LPCSTR szServer,
    // name
of SQL server
    LPCSTR szUser,
    // user name for login

```

```

        LPCSTR szPassword,
        //
password for login
    LPCSTR szHost,
    // not used
    LPCSTR szDatabase,
    // name
of database to use
    LPCWSTR szSPPrefix,
    // prefix
to append to the stored procedure names
    BOOL
bCallNoDuplicatesNewOrder //
whether to check for non-duplicate items in
NewOrder and call a new SP
)
:
m_bCallNoDuplicatesNewOrder(bCallNoDuplicates
NewOrder)
{
    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;

    wcsncpy(m_szSPPrefix, szSPPrefix,
sizeof(m_szSPPrefix)/sizeof(m_szSPPrefix[0]));

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

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

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

#ifdef COMPILE_FOR_SNAC

```

```

        sprintf( szConnectStr,
"DRIVER=SQL
Server;SERVER=%s;UID=%s;PWD=%s;DATABA
SE=%s",
        szServer, szUser, szPassword,
szDatabase );
    #else
        // Compile for SNAC
        sprintf( szConnectStr,
"DRIVER=SQL Native
Client;SERVER=%s;UID=%s;PWD=%s;DATABA
E=%s",
        szServer, szUser, szPassword,
szDatabase );
    #endif

    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::eAllocHand
le);

    {
        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::eExecDirec
t);

        // 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::eExecDirec
t);

```

```

        if (
SQLBindCol(m_hstmt, 1, SQL_C_CHAR,
&db_sp_version, sizeof(db_sp_version), NULL) !=
SQL_SUCCESS )
            ThrowError(CODBCERR::eBindCol);
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_LEN];
    H];
    char
    szTmp[6*SQL_MAX_MESSAGE_LEN];
    CODBCERR *pODBCerr;
    // not allocated until
    needed (maybe never)

    pODBCerr = new CODBCERR();
    pODBCerr->m_NativeError = 0;

```

```

        pODBCerr->m_bDeadLock = TRUE;
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 &&
            strstr(szMsg, sErrTimeoutExpired)
!= NULL))
                pODBCerr->m_bDeadLock = TRUE;
            // capture the (first)
            database error
            if (pODBCerr-
>m_NativeError == 0 && INativeError != 0)
                pODBCerr->m_NativeError =
INativeError;
            // quit if there isn't
            enough room to concatenate 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::eAllocHand
le);

        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::eBindPara
m);
        if ( SQLBindCol(m_hstmt, 1,
SQL_C_SLONG, &m_txn.StockLevel.low_stock, 0,
NULL) != SQL_SUCCESS )
            ThrowError(CODBCERR::eBindCol);
//Compose Stock Level statement
        _snwprintf(m_szStockLevelComman
d,
sizeof(m_szStockLevelCommand)/sizeof(m_szSto
ckLevelCommand[0]),
            L"{call
%stpcc_stocklevel (?,?,?)}", m_szSPPrefix);
}
void CTPCC_ODBC::StockLevel()
{
    RETCODE          rc;
    int
    iTryCount = 0;
    m_hstmt = m_hstmtStockLevel;
    while (TRUE)
    {
        try
        {
            rc =
SQLExecDirectW(m_hstmt,
m_szStockLevelCommand, SQL_NTS);
            if (rc !=
SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)
                ThrowError(CODBCERR::eExecDirec
t);
        }
        if (
SQLFetch(m_hstmt) == SQL_ERROR )
            ThrowError(CODBCERR::eFetch);

```

```

        SQLFreeStmt(m_hstmt,
SQL_CLOSE);

        m_txn.StockLevel.exec_status_code
= eOK;
                break;
        }
        catch (COBDCERR *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_RE
TRIED_TRANS, iTryCount);
}

void CTPCC_ODBC::InitNewOrderParams()
{
        if (
SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtNewOrder) != SQL_SUCCESS
                ||
SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtNewOrderNoDuplicates) !=
SQL_SUCCESS
                ||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderCols1) != SQL_SUCCESS
                ||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderCols2) != SQL_SUCCESS
                ||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderNoDuplicatesCols1) !=
SQL_SUCCESS
                ||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderNoDuplicatesCols2) !=
SQL_SUCCESS
        )
        ThrowError(COBCERR::eAllocHand
le);

        m_hstmt = m_hstmtNewOrder;

        if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descNewOrderCols1, SQL_IS_POINTER ) !=
SQL_SUCCESS )
        ThrowError(COBCERR::eSetStmtAt
tr);

        int i = 0;

```

```

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

        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(COBCERR::eBindPara
m);
        }

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

        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(COBCERR::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(COBCERR::eSetStmtAt
tr);

        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(COBCERR::eBindCol);

        //Compose the New Order
statement

```



```

        _snwprintf(m_szNewOrderComman
d,
sizeof(m_szNewOrderCommand)/sizeof(m_szNew
OrderCommand[0]),
// 0 1 2
//
012345678901234567890123456789
L"{call
%stpc_neworder(?,?,?,?,?,?,?,?,?,?,?,?,
?,?,?,?,?)",
L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,
?,?,?,?,?)", m_szSPPrefix);

m_iBeginNewOrderVariablePart =
29 + wcslen(m_szSPPrefix); // fixed
part + prefix part

////////////////////////////////////
////////////////////////////////////
//
// Now initialize New
Order that works on no duplicate (w_id,i_id)
pairs
// and returns one result
set for lineitem details.
//
//
m_hstmt =
m_hstmtNewOrderNoDuplicates;

if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descNewOrderNoDuplicatesCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )

ThrowError(CODBCERR::eSetStmtAt
tr);

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

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::eBindPara
m);
}
// set row-wise binding
if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_BIND_TYPE,
(SQLPOINTER)sizeof(m_txn.NewOrder.OL[0]),
SQL_IS_UIINTEGER) != SQL_SUCCESS
||
SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROWS_FETCHED_PTR,
&m_RowsFetched, 0) != SQL_SUCCESS )

ThrowError(CODBCERR::eSetStmtAt
tr);

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_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_descNewOrderNoDuplicatesCols2,
SQL_IS_POINTER ) != SQL_SUCCESS )

```

```

        ThrowError(CODBCERR::eSetStmtAt
tr);

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);

//Compose the New Order
statement
_snwprintf(m_szNewOrderNoDuplic
atesCommand,
sizeof(m_szNewOrderNoDuplicatesCommand)/siz
eof(m_szNewOrderNoDuplicatesCommand[0]),
L"{call
%stpc_neworder_new(?,?,?,?,?,?,?,?,?,
?,?,?,?,?)",
L"?,?,?,?,?,?,?,?,?,?,?,?,
?,?,?,?,?)", m_szSPPrefix);

m_iBeginNewOrderNoDuplicatesVari
ablePart = 33 + wcslen(m_szSPPrefix); // fixed
part + prefix part
}

//
// Returns true if there are duplicate
(warehouse_id, item_id)
// lineitem pairs in New Order input
parameters.
//
bool CTPCC_ODBC::DuplicatesInNewOrder()
{
    int i, j;

    for (i = 0; i <
m_txn.NewOrder.o_ol_cnt; ++i)
    {

```

```

        for (j = i+1; j <
m_bxn.NewOrder.o_ol_cnt; ++j)
        {
            if
(m_bxn.NewOrder.OL[i].ol_i_id ==
m_bxn.NewOrder.OL[j].ol_i_id)
            {
                return true;
            }
        }
        return false;
    }

void CTPCC_ODBC::NewOrder()
{
    if (m_bCallNoDuplicatesNewOrder)
    {
        if
(DuplicatesInNewOrder())
        {
            NewOrderDuplicates();
        }
        else
        {
            NewOrderNoDuplicates();
        }
    }
    else
    {
        NewOrderDuplicates();
    }
}

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

    // 0 1 2

    //
012345678901234567890123456789
wchar_t

    szSqlTemplate[iMAX_SP_NAME_LEN
];

    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::eSetStmtAt
tr);

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

        // check whether any order lines
are for a remote warehouse
        m_bxn.NewOrder.o_all_local = 1;
        for (i = 0; i <
m_bxn.NewOrder.o_ol_cnt; i++)
        {
            if
(m_bxn.NewOrder.OL[i].ol_supply_w_id !=
m_bxn.NewOrder.w_id)
            {
                m_bxn.NewOrder.o_all_local = 0;
                // at least one remote warehouse
                break;
            }
        }
        while (TRUE)
        {
            try
            {
                m_BindOffset = 0;
                rc =
SQLExecDirectW(m_hstmt, szSqlTemplate,
SQL_NTS);
                if (rc !=
SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)
                    ThrowError(CODBCERR::eExecDirec
t);

                // Get
order line results
                m_bxn.NewOrder.total_amount = 0;
                for (i =
0; i < m_bxn.NewOrder.o_ol_cnt; i++)
                {
                    // set the bind offset value...
                    m_BindOffset = i *
sizeof(m_bxn.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::eMoreResu
lts);

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

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

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

        SQLFreeStmt(m_hstmt,
SQL_CLOSE);

        if
(m_no_commit_flag == 1)
        {
            m_bxn.NewOrder.total_amount *=
((1 + m_bxn.NewOrder.w_tax +
m_bxn.NewOrder.d_tax) * (1 -
m_bxn.NewOrder.c_discount));
            m_bxn.NewOrder.exec_status_code
= eOK;
        }
        else
            m_bxn.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_RE
TRIED_TRANS, iTryCount);
}

//
// No lineitem duplicates optimized
// version.
//
void CTPCC_ODBC::NewOrderNoDuplicates()
{
    int
    RETCODE          i;
    int              rc;
    int              iTryCount = 0;

    // 0   1   2   3

    //
    0123456789012345678901234567890123
    wchar_t

    szSqlTemplate[iMAX_SP_NAME_LEN
];

L"{call tpcc_neworder_new(?,?,?,?," //
//
L"?,?,?,?,?,?,?,?,?,?,?," //
//
L"?,?,?,?,?,?,?,?,?,?,?," //
//
L"?,?,?,?,?,?,?,?,?,?,?}";
//
    m_hstmt =
m_hstmtNewOrderNoDuplicates;

    // associate the parameter and
    column bindings for this transaction
    if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descNewOrderNoDuplicatesCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )

        ThrowError(CODBCERR::eSetStmtAt
tr);

    // clip statement buffer based on
    number of parameters
    // fixed part is 33 chars and
    variable part is 6 chars per line item
    wcsncpy(szSqlTemplate,
m_szNewOrderNoDuplicatesCommand);

```

```

i =
m_iBeginNewOrderNoDuplicatesVariablePart +
m_bxn.NewOrder.ol_cnt; szSqlTemplate[i, L"");

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

while (TRUE)
{
    try
    {
        //
        configure block cursor
        if (
SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_ARRAY_SIZE, (SQLPOINTER)1,
0) != SQL_SUCCESS )

            ThrowError(CODBCERR::eSetStmtAt
tr);

        rc =
SQLExecDirectW(m_hstmt, szSqlTemplate,
SQL_NTS);

        if (rc !=
SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

            ThrowError(CODBCERR::eExecDirec
t);

        //
        configure block cursor
        if
(SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)MAX_OL_NEW_ORDER_ITEMS, 0)
!= SQL_SUCCESS)

            ThrowError(CODBCERR::eSetStmtAt
tr);

        // Get
        order line results
        if (
SQLFetch(m_hstmt) == SQL_ERROR)

            ThrowError(CODBCERR::eFetch);

        m_bxn.NewOrder.total_amount = 0;
        for (i =
0; i < m_bxn.NewOrder.o_ol_cnt; i++)
        {

```

```

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

//
associate the column bindings for the second
result set
if (
SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descNewOrderNoDuplicatesCols2,
SQL_IS_POINTER ) != SQL_SUCCESS )

    ThrowError(CODBCERR::eSetStmtAt
tr);

// move
to the next resultset
if (
SQLMoreResults(m_hstmt) == SQL_ERROR )

    ThrowError(CODBCERR::eMoreResu
lts);

if (
SQLFetch(m_hstmt) == SQL_ERROR)

    ThrowError(CODBCERR::eFetch);

SQLFreeStmt(m_hstmt,
SQL_CLOSE);

if
(m_no_commit_flag == 1)
{
    m_bxn.NewOrder.total_amount *=
((1 + m_bxn.NewOrder.w_tax +
m_bxn.NewOrder.d_tax) * (1 -
m_bxn.NewOrder.c_discount));

    m_bxn.NewOrder.exec_status_code
= eOK;
}
else
    m_bxn.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_RE
TRIED_TRANS, iTryCount);
}

```

```

void CTPCC_ODBC::InitPaymentParams()
{
    if (
        SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
            &m_hstmtPayment) != SQL_SUCCESS )
        ThrowError(CODBCERR::eAllocHand
le);

    m_hstmt = m_hstmtPayment;

    int i = 0;
    if ( SQLBindParameter(m_hstmt,
        ++i, SQL_PARAM_INPUT, SQL_C_SLONG,
        SQL_INTEGER, 0, 0, &m_bxn.Payment.w_id, 0,
        NULL) != SQL_SUCCESS
        ||
        SQLBindParameter(m_hstmt, ++i,
        SQL_PARAM_INPUT, SQL_C_SLONG,
        SQL_INTEGER, 0, 0, &m_bxn.Payment.c_w_id, 0,
        NULL) != SQL_SUCCESS
        ||
        SQLBindParameter(m_hstmt, ++i,
        SQL_PARAM_INPUT, SQL_C_DOUBLE,
        SQL_NUMERIC, 6, 2,
        &m_bxn.Payment.h_amount, 0, NULL) !=
        SQL_SUCCESS
        ||
        SQLBindParameter(m_hstmt, ++i,
        SQL_PARAM_INPUT, SQL_C_UTINYINT,
        SQL_TINYINT, 0, 0, &m_bxn.Payment.d_id, 0,
        NULL) != SQL_SUCCESS
        ||
        SQLBindParameter(m_hstmt, ++i,
        SQL_PARAM_INPUT, SQL_C_UTINYINT,
        SQL_TINYINT, 0, 0, &m_bxn.Payment.c_d_id, 0,
        NULL) != SQL_SUCCESS
        ||
        SQLBindParameter(m_hstmt, ++i,
        SQL_PARAM_INPUT, SQL_C_SLONG,
        SQL_INTEGER, 0, 0, &m_bxn.Payment.c_id, 0,
        NULL) != SQL_SUCCESS
        ||
        SQLBindParameter(m_hstmt, ++i,
        SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
        sizeof(m_bxn.Payment.c_last), 0,
        &m_bxn.Payment.c_last,
        sizeof(m_bxn.Payment.c_last), NULL) !=
        SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindPara
m);

    i = 0;
    if ( SQLBindCol(m_hstmt, ++i,
        SQL_C_SLONG, &m_bxn.Payment.c_id,
        0, NULL) !=
        SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.c_last,
        sizeof(m_bxn.Payment.c_last),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i,
        SQL_C_TYPE_TIMESTAMP,
        &m_bxn.Payment.h_date, 0, NULL)
        != SQL_SUCCESS

```

```

        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.w_street_1,
        sizeof(m_bxn.Payment.w_street_1),
        NULL) != SQL_SUCCESS ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.w_street_2,
        sizeof(m_bxn.Payment.w_street_2),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.w_city,
        sizeof(m_bxn.Payment.w_city),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.w_state,
        sizeof(m_bxn.Payment.w_state),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.w_zip,
        sizeof(m_bxn.Payment.w_zip),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.d_street_1,
        sizeof(m_bxn.Payment.d_street_1),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.d_street_2,
        sizeof(m_bxn.Payment.d_street_2),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.d_city,
        sizeof(m_bxn.Payment.d_city),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.d_state,
        sizeof(m_bxn.Payment.d_state),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.d_zip,
        sizeof(m_bxn.Payment.d_zip),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.c_first,
        sizeof(m_bxn.Payment.c_first),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.c_middle,
        sizeof(m_bxn.Payment.c_middle),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.c_street_1,
        sizeof(m_bxn.Payment.c_street_1),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.c_street_2,
        sizeof(m_bxn.Payment.c_street_2),
        NULL) != SQL_SUCCESS

```

```

        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.c_city,
        sizeof(m_bxn.Payment.c_city),
        NULL) != SQL_SUCCESS ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.c_state,
        sizeof(m_bxn.Payment.c_state),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.c_zip,
        sizeof(m_bxn.Payment.c_zip),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.c_phone,
        sizeof(m_bxn.Payment.c_phone),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i,
        SQL_C_TYPE_TIMESTAMP,
        &m_bxn.Payment.c_since, 0, NULL)
        != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.c_credit,
        sizeof(m_bxn.Payment.c_credit),
        NULL) != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
        &m_bxn.Payment.c_credit_lim, 0, NULL) !=
        SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
        &m_bxn.Payment.c_discount, 0, NULL)
        != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
        &m_bxn.Payment.c_balance, 0, NULL)
        != SQL_SUCCESS
        ||
        SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
        &m_bxn.Payment.c_data,
        sizeof(m_bxn.Payment.c_data),
        NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindCol);

    //Compose Payment statement

    _snwprintf(m_szPaymentCommand,
        sizeof(m_szPaymentCommand)/sizeof(m_szPaym
entCommand[0]),
        L"{call
        %stpc_payment(?,?,?,?)}", m_szSPPrefix);
}

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

    m_hstmt = m_hstmtPayment;

    if (m_bxn.Payment.c_id != 0)

        m_bxn.Payment.c_last[0] = 0;

    while (TRUE)

```

```

    {
        try
        {
            rc =
SQLExecDirectW(m_hstmt,
m_szPaymentCommand, SQL_NTS);
            if (rc !=
SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)
                ThrowError(CODBCERR::eExecDirec
t);
            if (
SQLFetch(m_hstmt) == SQL_ERROR)
                ThrowError(CODBCERR::eFetch);
            SQLFreeStmt(m_hstmt,
SQL_CLOSE);
            if
(m_bxn.Payment.c_id == 0)
                throw new CTPCC_ODBC_ERR(
CTPCC_ODBC_ERR::ERR_INVALID_CUST );
            else
                m_bxn.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_RE
TRIED_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::eAllocHand
le);
    m_hstmt = m_hstmtOrderStatus;

```

```

        if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols1, SQL_IS_POINTER ) !=
SQL_SUCCESS )
            ThrowError(CODBCERR::eSetStmtAt
tr);
        int i = 0;
        if ( SQLBindParameter(m_hstmt,
++i, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_bxn.OrderStatus.w_id,
0, NULL) != SQL_SUCCESS
            ||
SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT,
SQL_TINYINT, 0, 0, &m_bxn.OrderStatus.d_id, 0,
NULL) != SQL_SUCCESS
            ||
SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_bxn.OrderStatus.c_id, 0,
NULL) != SQL_SUCCESS
            ||
SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
sizeof(m_bxn.OrderStatus.c_last), 0,
&m_bxn.OrderStatus.c_last,
sizeof(m_bxn.OrderStatus.c_last), NULL) !=
SQL_SUCCESS
        )
            ThrowError(CODBCERR::eBindPara
m);
        // configure block cursor
        if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_BIND_TYPE,
(SQLPOINTER)sizeof(m_bxn.OrderStatus.OL[0]),
0) != SQL_SUCCESS
            ||
SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROWS_FETCHED_PTR,
&m_RowsFetched, 0) != SQL_SUCCESS
        )
            ThrowError(CODBCERR::eSetStmtAt
tr);
        i = 0;
        if ( SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG,
&m_bxn.OrderStatus.OL[0].ol_supply_w_id, 0,
NULL) != SQL_SUCCESS
            ||
SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_bxn.OrderStatus.OL[0].ol_i_id, 0, NULL) !=
SQL_SUCCESS
            ||
SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_bxn.OrderStatus.OL[0].ol_quantity, 0, NULL)
!= SQL_SUCCESS
            ||
SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_bxn.OrderStatus.OL[0].ol_amount, 0, NULL)
!= SQL_SUCCESS
            ||
SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP,
&m_bxn.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::eSetStmtAt
tr);
        i = 0;
        if ( SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_bxn.OrderStatus.c_id, 0,
NULL) != SQL_SUCCESS
            ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.OrderStatus.c_last,
sizeof(m_bxn.OrderStatus.c_last), NULL) !=
SQL_SUCCESS
            ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.OrderStatus.c_first,
sizeof(m_bxn.OrderStatus.c_first), NULL) !=
SQL_SUCCESS
            ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.OrderStatus.c_middle,
sizeof(m_bxn.OrderStatus.c_middle), NULL) !=
SQL_SUCCESS
            ||
SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP,
&m_bxn.OrderStatus.o_entry_d, 0, NULL) !=
SQL_SUCCESS
            ||
SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_bxn.OrderStatus.o_carrier_id, 0, NULL) !=
SQL_SUCCESS
            ||
SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_bxn.OrderStatus.c_balance, 0, NULL) !=
SQL_SUCCESS
            ||
SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_bxn.OrderStatus.o_id, 0, NULL) !=
SQL_SUCCESS
        )
            ThrowError(CODBCERR::eBindCol);
        //Compose Order Status statement
        _snwprintf(m_szOrderStatusComma
nd,
sizeof(m_szOrderStatusCommand)/sizeof(m_szOr
derStatusCommand[0]),
        L"(call
%stpc_orderstatus(?,?,?,?)", m_szSPPrefix);
    }
    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::eSetStmtAt
tr);

        if (m_bxn.OrderStatus.c_id != 0)

        m_bxn.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::eSetStmtAt
tr);

                rc =
SQLExecDirectW(m_hstmt,
m_szOrderStatusCommand, SQL_NTS);
                if ((rc
== SQL_SUCCESS_WITH_INFO) &&
(m_RowsFetched != 0) || (rc == SQL_ERROR) )

                ThrowError(CODBCERR::eExecDirec
t);

                //
                configure block cursor
                if (
SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)MAX_OL_ORDER_STATUS_ITEMS,
0) != SQL_SUCCESS )

                ThrowError(CODBCERR::eSetStmtAt
tr);

                rc =
SQLFetchScroll( m_hstmt, SQL_FETCH_NEXT, 0
);
                if ((rc
== SQL_SUCCESS_WITH_INFO) &&
(m_RowsFetched != 0) || (rc == SQL_ERROR) )

                ThrowError(CODBCERR::eFetchScro
ll);

                m_bxn.OrderStatus.o_ol_cnt =
(short)m_RowsFetched;

                if
(m_bxn.OrderStatus.o_ol_cnt != 0)
                {
                    if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols2, SQL_IS_POINTER ) !=
SQL_SUCCESS )

                    ThrowError(CODBCERR::eSetStmtAt
tr);

```

```

        if ( SQLMoreResults(m_hstmt) ==
SQL_ERROR )

        ThrowError(CODBCERR::eMoreResu
lts);

        if ( (rc = SQLFetch(m_hstmt)) ==
SQL_ERROR)

        ThrowError(CODBCERR::eFetch);

        SQLFreeStmt(m_hstmt,
SQL_CLOSE);

        if
(m_bxn.OrderStatus.o_ol_cnt == 0)

        throw new CTPCC_ODBC_ERR(
CTPCC_ODBC_ERR::ERR_NO_SUCH_ORDER );
        else if
(m_bxn.OrderStatus.c_id == 0 &&
m_bxn.OrderStatus.c_last[0] == 0)

        throw new CTPCC_ODBC_ERR(
CTPCC_ODBC_ERR::ERR_INVALID_CUST );
        else

        m_bxn.OrderStatus.exec_status_cod
e = 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_RE
TRIED_TRANS, iTryCount);
}

void CTPCC_ODBC::InitDeliveryParams()
{
    if (
SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtDelivery) != SQL_SUCCESS )

    ThrowError(CODBCERR::eAllocHand
le);

    m_hstmt = m_hstmtDelivery;

    int i = 0;

```

```

        if ( SQLBindParameter(m_hstmt,
++, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_bxn.Delivery.w_id, 0,
NULL) != SQL_SUCCESS ||
SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0,
&m_bxn.Delivery.o_carrier_id, 0, NULL) !=
SQL_SUCCESS
        )

        ThrowError(CODBCERR::eBindPara
m);

        for (i=0;i<10;i++)
        {
            if (
SQLBindCol(m_hstmt, (UWORD)(i+1),
SQL_C_SLONG, &m_bxn.Delivery.o_id[i], 0,
NULL) != SQL_SUCCESS )

            ThrowError(CODBCERR::eBindCol);
        }

        //Compose Delivery statement
        _snprintf(m_szDeliveryCommand,
sizeof(m_szDeliveryCommand)/sizeof(m_szDelive
ryCommand[0]),
            L"{call
%stpc_delivery (?,?,?)", m_szSPPrefix);
    }

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

        m_hstmt = m_hstmtDelivery;

        while (TRUE)
        {
            try
            {
                rc =
SQLExecDirectW(m_hstmt,
m_szDeliveryCommand, SQL_NTS);
                if (rc !=
SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

                ThrowError(CODBCERR::eExecDirec
t);

                if (
SQLFetch(m_hstmt) == SQL_ERROR )

                ThrowError(CODBCERR::eFetch);

                SQLFreeStmt(m_hstmt,
SQL_CLOSE);

                m_bxn.Delivery.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)
    }
}
// if (iTryCount)
// throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RE
TRIED_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

#define iMAX_SP_NAME_LEN 256
//maximum length of a stored
procedure name with parameters

class CODBCERR : 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;};

char*
ErrorTypeStr() { return "ODBC"; }

int
ErrorNum() {return
m_NativeError;};

```

```

char*
ErrorText() {return m_odbcerrstr;};
ErrorAction() { return
(int)m_eAction; }
};

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;};

char*
ErrorTypeStr() { return "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;
        SQLHSTMT
m_hstmt;
// the current hstmt

        SQLHSTMT
m_hstmtNewOrder;
        SQLHSTMT
m_hstmtNewOrderNoDuplicates;
// NewOrder with one result set for
lineitem details
        SQLHSTMT
m_hstmtPayment;
        SQLHSTMT
m_hstmtDelivery;
        SQLHSTMT
m_hstmtOrderStatus;
        SQLHSTMT
m_hstmtStockLevel;

        SQLHDESC
m_descNewOrderCols1;
        SQLHDESC
m_descNewOrderCols2;
        SQLHDESC
m_descNewOrderNoDuplicatesCols1
;
// NewOrder with one result set for
lineitem details
        SQLHDESC
m_descNewOrderNoDuplicatesCols2
;
// NewOrder with one result set for
lineitem details
        SQLHDESC
m_descOrderStatusCols1;
        SQLHDESC
m_descOrderStatusCols2;

        wchar_t
m_szSPPrefix[32]; // stored
procedures prefix

        wchar_t
m_szNewOrderCommand[iMAX_SP_
NAME_LEN];
        wchar_t
m_szNewOrderNoDuplicatesComma
nd[iMAX_SP_NAME_LEN];
        int
m_iBeginNewOrderVariablePart;
// beginning of the variable part in
NewOrder statement
        int
m_iBeginNewOrderNoDuplicatesVari
ablePart; // beginning of the variable part in
NewOrder statement
        wchar_t
m_szPaymentCommand[iMAX_SP_N
AME_LEN];
        wchar_t
m_szDeliveryCommand[iMAX_SP_N
AME_LEN];
        wchar_t
m_szOrderStatusCommand[iMAX_S
P_NAME_LEN];
        wchar_t
m_szStockLevelCommand[iMAX_SP
_NAME_LEN];

// new-order specific
fields

```

```

        SQLINTEGER
m_BindOffset;
        SQLINTEGER
m_RowsFetched;
        int
m_no_commit_flag;
// tpcc_neworder_new
flag
        BOOL
m_bCallNoDuplicatesNewOrder;

        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;

        bool
DuplicatesInNewOrder();
        void
NewOrderDuplicates();
        void
NewOrderNoDuplicates();

public:
        CTPCC_ODBC(
        LPCSTR szServer, LPCSTR szUser,
        LPCSTR szPassword,
        LPCSTR szHost,
        LPCSTR szDatabase,
        LPCWSTR szSPPrefix,
        BOOL bCallNoDuplicatesNewOrder);
~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; };
PSTOCK_LEVEL_DATA
BuffAddr_StockLevel()
{ return &m_txn.StockLevel; };
        inline
PORORDER_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,
        LPCWSTR szSPPrefix,
        BOOL bCallNoDuplicatesNewOrder );

typedef CTPCC_ODBC*
(TYPE_CTPCC_ODBC)(LPCSTR, LPCSTR, LPCSTR,
LPCSTR, LPCSTR, LPCWSTR, BOOL);

```

**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.
#ifndef DllDecl

```



```

#define DllDecl __declspec( dllimport )
#endif

class CCOMERR : public CBaseErr
{
private:
    char
    m_szErrorText[64];

public:
    // use this interface
    for genuine COM errors
    CCOMERR( HRESULT
    hr )
    {
        m_hr =
        hr;

        m_iErrorType = 0;
        m_iError
        = 0;
    }

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

    int
    m_hr;
    int
    m_iErrorType;
    int
    m_iError;

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

    char *ErrorTypeStr() {
    return "COM"; }

    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;
    };

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

    inline
    PNEW_ORDER_DATA
    BuffAddr_NewOrder()
    { return &m_pTnx->u.NewOrder;
    };

    inline
    PPAYMENT_DATA
    BuffAddr_Payment()
    { return &m_pTnx->u.Payment;
    };

    inline
    PDELIVERY_DATA
    BuffAddr_Delivery()
    { return &m_pTnx->u.Delivery; };
};

```

```

        inline
        PSTOCK_LEVEL_DATA
        BuffAddr_StockLevel()
        { return &m_pTnx->u.StockLevel;
        };
        inline
        PORDER_STATUS_DATA
        BuffAddr_OrderStatus() { return
        &m_pTnx->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(dlllexport) CTPCC_COM*
CTPCC_COM_new(BOOL);

typedef CTPCC_COM*
(TYPE_CTPCC_COM)(BOOL);

db_oledb_dll\tpcc_oledb.h

/* FILE:
TPCC_OLEDB.H
*
* Microsoft TPC-C Kit Ver. 4.20.000
*
* Copyright Microsoft, 1999-2004
*
* Written by Sergey Vasilevskiy
* All Rights Reserved
*
*
* PURPOSE: Header file for TPC-C
txn class OLE DB implementation.
*
*
*/
#pragma once

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

```

```

#define iMAX_SP_NAME_LEN 256
//maximum length of a stored
procedure name with parameters

// Type of parameter and result set column
bindings.
enum eBindingType
{
    eInputParameter,
    eOutputParameter,
    eInputOutputParameter,
    eOutputColumn
};

class COLEDBERR : public CBaseErr
{
public:
    enum ACTION
    {
        eNone,

        eUnknown,

        eQueryInterface,
        // error from
        QueryInterface

        eCreateSession,

        eCreateCommand,

        eSetCommandText,

        eExecute,
        // = 6

        eCreateAccessor,

        ePrepare,

        eGetNextRows,

        eGetData,

        eGetResult
        // = 11
    };

    COLEDBERR(LPCTSTR
szLoc)
    :
    CBaseErr(szLoc)
    {
        m_eAction = eNone;

        m_NativeError = 0;

        m_bDeadLock = FALSE;

        m_OLEDBErrStr = NULL;
    };

    ~COLEDBERR()
    {
        if
(m_OLEDBErrStr != NULL)

        delete [] m_OLEDBErrStr;
    };
};

```

```

        ACTION
        m_eAction;
        int
        m_NativeError;
        BOOL
        m_bDeadLock;
        char
        *m_OLEDBErrStr;

        int
        ErrorType() {return
ERR_TYPE_OLEDB;};

        char*
        ErrorTypeStr() { return "OLEDB"; }

        int
        ErrorNum() {return
m_NativeError;};

        char*
        ErrorText() {return
m_OLEDBErrStr;};

        int
        ErrorAction() { return
(int)m_eAction; }
};

class CTPCC_OLEDB_ERR : public CBaseErr
{
public:
    enum
    TPCC_OLEDB_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_OLEDB_ERR(
int iErr ) { m_erno = iErr; m_iTryCount = 0; };

    CTPCC_OLEDB_ERR(
int iErr, int iTryCount ) { m_erno = iErr;
m_iTryCount = iTryCount; };

        int
        m_erno;
        int
        m_iTryCount;

        int
        ErrorType() {return
ERR_TYPE_TPCC_OLEDB;};

        char*
        ErrorTypeStr() { return "TPCC
OLEDB"; }

        int
        ErrorNum() {return m_erno;};

        char*
        ErrorText();
};

```

```

class DllDecl CTPCC_OLEDB : 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

        DBPROPSET

        m_rgInitPropSet;
// initialization property set used to
establish a connection
        DBPROP

        m_InitProperties[4]; //
individual initialization properties

        IDBCreateSession*

        m_pIDBCreateSession; //
session (connection) interface
        IDBCreateCommand*

        m_pIDBCreateCommand; // SQL
command creation interface

        IMalloc*

        m_pIMalloc;
// Needed to release
error strings.

        // StockLevel
ICommandText*

        m_pIStockLevelCommand;
HACCESSOR

        m_hStockLevelInputAccessor;
// accessor to bind input
parameters
        HACCESSOR

        m_hStockLevelOutputAccessor;
// accessor to bind output columns
DBPARAMS

        m_StockLevelExecuteParams;
// parameter structure
for Execute

        // NewOrder
// One prepared
command for each possible number of new order
line items
        ICommandText*

        m_pINewOrderCommand[MAX_OL_
NEW_ORDER_ITEMS];
// accessors to bind
input parameters
// one for each
possible number of new order line items

```

```

HACCESSOR

m_hNewOrderInputAccessor[MAX_
OL_NEW_ORDER_ITEMS];
// accessor to bind
output columns of the first rowset
HACCESSOR

m_hNewOrderOutputAccessor[MAX_
OL_NEW_ORDER_ITEMS];
// accessor to bind
output columns of the second rowset
HACCESSOR

m_hNewOrderOutputAccessor2[MA
X_OL_NEW_ORDER_ITEMS];
// parameter structure
for Execute
DBPARAMS

m_NewOrderExecuteParams[MAX_
OL_NEW_ORDER_ITEMS];
// Payment
ICommandText*

m_pIPaymentCommand;
HACCESSOR

m_hPaymentInputAccessor;
// accessor to bind input
parameters
HACCESSOR

m_hPaymentOutputAccessor;
// accessor to bind output columns
DBPARAMS

m_PaymentExecuteParams;
// parameter structure
for Execute

// OrderStatus
ICommandText*

m_pIOrderStatusCommand;
HACCESSOR

m_hOrderStatusInputAccessor;
// accessor to bind input
parameters
HACCESSOR

m_hOrderStatusOutputAccessor;
// accessor to bind output columns
HACCESSOR

m_hOrderStatusOutputAccessor2;
// accessor to bind output columns
DBPARAMS

m_OrderStatusExecuteParams;
// parameter structure
for Execute

// Delivery
ICommandText*

m_pIDeliveryCommand;
HACCESSOR

m_hDeliveryInputAccessor;
// accessor to bind input
parameters

```

```

HACCESSOR

m_hDeliveryOutputAccessor;
// accessor to bind output columns
DBPARAMS

m_DeliveryExecuteParams;//
parameter structure for Execute

wchar_t

m_szSPPrefix[32]; // stored
procedures prefix

// new-order specific
fields
int

m_no_commit_flag;

void ThrowError(
IUnknown* pObjectWithError,
COLEDBERR::ACTION eAction, LPCWSTR
szLocation );

void
CheckSPVersion();

void
InitNewOrderParams();

void
InitPaymentParams();

void
InitDeliveryParams();

void
InitStockLevelParams();

void
InitOrderStatusParams();

// Helper function to
create and prepare a command
void
CreateCommand(wchar_t* szSQLCommand,
ICommandText** ppICommandText);
// Helper function to
prepare a command
void
PrepareCommand(ICommandText* pICommand);

// Helper function to
fill one binding
// Used for both input
parameter and output column bindings
void
SetBinding(DBBINDING* pDBBinding, size_t
obValue, size_t cbMaxLen, DBTYPE wType);

// Helper function to
initialize an array of bindings
void
InitBindings(DBBINDING* pDBBindings, int
iCount, eBindingType BindingType);

union
{
NEW_ORDER_DATA
NewOrder;

PAYMENT_DATA
Payment;

DELIVERY_DATA
Delivery;

```

```

STOCK_LEVEL_DATA
StockLevel;
ORDER_STATUS_DATA
OrderStatus;
}
m_txn;

public:
CTPCC_OLEDB(LPCWSTR szServer,
LPCWSTR szUser, LPCWSTR szPassword, LPCWSTR
szHost, LPCWSTR szDatabase, LPCWSTR
szSPPrefix);

~CTPCC_OLEDB(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_OLEDB*
CTPCC_OLEDB_new
( LPCWSTR szServer, LPCWSTR szUser,
LPCWSTR szPassword, LPCWSTR szHost, LPCWSTR
szDatabase, LPCWSTR szSPPrefix );

typedef CTPCC_OLEDB*
(TYPE_CTPCC_OLEDB)(LPCWSTR, LPCWSTR,
LPCWSTR, LPCWSTR, LPCWSTR, LPCWSTR);

```

```

db oledb dll\src\tpcc oledb
.cpp

```

```

/* FILE:
TPCC_OLEDB.CPP
*
* Microsoft TPC-C Kit Ver. 4.42.000
*
* Copyright Microsoft, 2004
*
* Written by Sergey Vasilevskiy
* All Rights Reserved
*
*
*
* PURPOSE: Implements OLEDB
calls for TPC-C bns.
* Contact: Charles Levine
(clevine@microsoft.com)
*
*/

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

#define DBINITCONSTANTS
#include <oledb.h>
#include <sqloledb.h> // Use MDAC
#include <sqlncli.h> // Use SNAC
#include <oledberr.h>

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

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

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

#ifdef SQL_MAX_MESSAGE_LENGTH
#define SQL_MAX_MESSAGE_LENGTH 512
#endif

// version string; must match return value from
tpcc_version stored proc
const char sVersion[] = "4.20.000";

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

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

// this needs to be the same as the max length of
machine/database/user/password in Benchcraft
(engstut.h)
const static int iMaxNameLen = 32;

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

```

```

case
DLL_PROCESS_DETACH:
break;

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

/* FUNCTION: CTPCC_OLEDB_ERR::ErrorText
*
*/
char* CTPCC_OLEDB_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,
"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_OLEDB*
CTPCC_OLEDB_new(
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
LPCWSTR szSPPrefix )
// prefix to append to the stored
procedure names
{
return new CTPCC_OLEDB(
szServer, szUser, szPassword, szHost,
szDatabase, szSPPrefix );
}

CTPCC_OLEDB::CTPCC_OLEDB (
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
LPCWSTR szSPPrefix
// prefix
to append to the stored procedure names
)
: m_pIMalloc(NULL)
{
int iRc;

int i;

HRESULT hr;
IDBInitialize*
pIDBInitialize = NULL;
// data source interface
IDBProperties*
pIDBProperties = NULL;
ICommandText*
pICommandText;
// SQL
command without parameters
wchar_t
szwServer[iMaxNameLen];
// Unicode string used to convert to
BSTR
wchar_t
szwDatabase[iMaxNameLen];
// Unicode string used to convert to
BSTR
wchar_t
szwUser[iMaxNameLen];
// Unicode string used to convert to
BSTR
wchar_t
szwPassword[iMaxNameLen];
// Unicode string used to convert to
BSTR

```

```

        // Copy stored procedures prefix
        wcsncpy(m_szSPPrefix, szSPPrefix,
        sizeof(m_szSPPrefix)/sizeof(m_szSPPrefix[0]));

        // Convert single byte ANSI strings
        to Unicode (for later conversion to BSTR)
        iRc =
        MultiByteToWideChar(CP_THREAD_ACP,
        MB_PRECOMPOSED, szServer,
        (int)strlen(szServer)+1, szwServer,
        iMaxNameLen);
        iRc =
        MultiByteToWideChar(CP_THREAD_ACP,
        MB_PRECOMPOSED, szDatabase,
        (int)strlen(szDatabase)+1, szwDatabase,
        iMaxNameLen);
        iRc =
        MultiByteToWideChar(CP_THREAD_ACP,
        MB_PRECOMPOSED, szUser,
        (int)strlen(szUser)+1, szwUser, iMaxNameLen);
        iRc =
        MultiByteToWideChar(CP_THREAD_ACP,
        MB_PRECOMPOSED, szPassword,
        (int)strlen(szPassword)+1, szwPassword,
        iMaxNameLen);

        // Initialize COM library to be able
        to use OLE-DB interfaces
        CoInitialize(NULL);

        // Initialization - create SQLOLEDB
        component
        //hr =
        CoCreateInstance(CLSID_SQLOLEDB, // GUID of
        SQLOLEDB component
        // Compile for SNAC
        hr =
        CoCreateInstance(CLSID_SQLNCLI, // GUID
        of SQLNCLI component
        NULL,
        // not defining an
        aggregate component, so NULL
        CLSCTX_INPROC_SERVER,
        // run the component in our
        process
        IID_IDBInitialize,
        (void **) &pIDBInitialize);
        /*
        Initialize the property values needed
        to establish the connection.
        */
        for(i = 0; i < 4; i++)
            VariantInit(&m_InitProperties[i].vValue);
        //Server name.
        m_InitProperties[0].dwPropertyID =
        DBPROP_INIT_DATASOURCE;
        m_InitProperties[0].vValue.vt = VT_BSTR;
        m_InitProperties[0].vValue.bstrVal=
        SysAllocString(szwServer);
        m_InitProperties[0].dwOptions =
        DBPROPOPTIONS_REQUIRED;
        m_InitProperties[0].colid = DB_NULLID;
        //Database.
        m_InitProperties[1].dwPropertyID =
        DBPROP_INIT_CATALOG;
        m_InitProperties[1].vValue.vt = VT_BSTR;
        m_InitProperties[1].vValue.bstrVal=
        SysAllocString(szwDatabase);
        m_InitProperties[1].dwOptions =
        DBPROPOPTIONS_REQUIRED;
        m_InitProperties[1].colid = DB_NULLID;
        //Username (login).
        m_InitProperties[2].dwPropertyID =
        DBPROP_AUTH_USERID;
        m_InitProperties[2].vValue.vt = VT_BSTR;

```

```

        m_InitProperties[2].vValue.bstrVal=
        SysAllocString(szwUser);
        m_InitProperties[2].dwOptions =
        DBPROPOPTIONS_REQUIRED;
        m_InitProperties[2].colid = DB_NULLID;
        //Password.
        m_InitProperties[3].dwPropertyID =
        DBPROP_AUTH_PASSWORD;

        m_InitProperties[3].vValue.vt = VT_BSTR;
        m_InitProperties[3].vValue.bstrVal=
        SysAllocString(szwPassword);
        m_InitProperties[3].dwOptions =
        DBPROPOPTIONS_REQUIRED;
        m_InitProperties[3].colid = DB_NULLID;
        /*
        Construct the DBPROPSSET
        structure(m_rgInitPropSet). The
        DBPROPSSET structure is used to pass an array
        of DBPROP
        structures (m_InitProperties) to the
        SetProperties method.
        */
        m_rgInitPropSet.guidPropertySet =
        DBPROPSET_DBINIT;
        m_rgInitPropSet.cProperties = 4;
        m_rgInitPropSet.rgProperties =
        m_InitProperties;
        //Set initialization properties.
        if (FAILED(hr = pIDBInitialize-
        >QueryInterface(IID_IDBProperties,
        (void
        **)&pIDBProperties)))
        {
            ThrowError(pIDBInitialize,
            COLEDBERR::eQueryInterface,
            "CTPCC_OLEDB()");
        }

        hr = pIDBProperties->SetProperties(1,
        &m_rgInitPropSet);

        pIDBProperties->Release();
        //Now establish the connection to the data
        source.
        hr = pIDBInitialize->Initialize();

        // Free BSTR property strings
        for(i = 0; i < 4; i++)
        {
            SysFreeString(m_InitProperties[i].vValue.bstrVal)
        }

        hr = pIDBInitialize-
        >QueryInterface(IID_IDBCreateSession, (void
        **)&m_pIDBCreateSession);

        // Releasing this has no effect on
        the SQL Server connection
        // of the data source object because of the
        reference maintained by
        // m_pIDBCreateSession.
        pIDBInitialize->Release();
        pIDBInitialize = NULL;

        hr = m_pIDBCreateSession-
        >CreateSession(NULL, IID_IDBCreateCommand,
        (IUnknown **) &m_pIDBCreateCommand);
        if (FAILED(hr))

```

```

        {
            ThrowError(m_pIDBCreateSession,
            COLEDBERR::eCreateSession,
            "CTPCC_OLEDB()");
        }

        hr = m_pIDBCreateCommand-
        >CreateCommand(NULL, IID_ICommandText,
        (IUnknown **) &pICommandText);

        if (FAILED(hr))
        {
            ThrowError(m_pIDBCreateCommand,
            COLEDBERR::eCreateCommand,
            "CTPCC_OLEDB()");
        }

        hr = pICommandText-
        >SetCommandText(DBGUID_SQL, L"set nocount
        on set XACT_ABORT ON");
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
            COLEDBERR::eSetCommandText,
            "CTPCC_OLEDB()");
        }

        hr = pICommandText-
        >Execute(NULL, IID_NULL, NULL, NULL, NULL);
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
            COLEDBERR::eExecute, "CTPCC_OLEDB()");
        }

        pICommandText->Release();

        // verify that version of stored procs
        on server is correct
        CheckSPVersion();

        // Get IMalloc interface
        hr = CoGetMalloc(1, (LPMMALLOC
        *) &m_pIMalloc);

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

    CTPCC_OLEDB::~~CTPCC_OLEDB( void )
    {
        if (m_pIMalloc != NULL)
        {
            m_pIMalloc-
            >Release();
        }
        m_pIPaymentCommand-
        >Release();
        m_pIDBCreateCommand-
        >Release();
        m_pIDBCreateSession->Release();

        CoUninitialize(); //
        uninitialize COM library
    }

```

```

/*
 *      Check stored procedures version on
the server.
 */
void CTPCC_OLEDB::CheckSPVersion()
{
    HRESULT
    hr;
    char
    db_sp_version[10];
    ICommandText*
    pICommandText;
    IAccessor*
    pIAccessor;
    IRowset*
    pRowset;
    const ULONG
    nOutputParams = 1;    // output
1st result set columns
    HACCESSOR
    hTpccVersionOutputAccessor;
    // Structure to bind in accessor
    DBBINDING
    acOutputDBBinding[nOutputParams
];
    DBBINDSTATUS
    acOutputDBBindStatus[nOutputPara
ms];
    LONG
    cRows = 1; // number of rows
returned in the rowset
    ULONG
    cRowsObtained;
    HROW
    rghRow;
    //returned row handles
    HROW*
    prghRow = &rghRow;

    hr = m_pIDBCreateCommand-
>CreateCommand(NULL, IID_ICommandText,
(IUnknown **)&pICommandText);
    if (FAILED(hr))
    {
        ThrowError(m_pIDBCreateComman
d, COLEDBERR::eCreateCommand,
"CheckSPVersion()");
    }

    hr = pICommandText-
>SetCommandText(DBGUID_SQL, L"{call
tpcc_version}");
    if (FAILED(hr))
    {
        ThrowError(pICommandText,
COLEDBERR::eSetCommandText,
"CheckSPVersion()");
    }

    hr = pICommandText-
>QueryInterface(IID_IAccessor, (void
**)&pIAccessor);
    if (FAILED(hr))
    {
        ThrowError(pICommandText,
COLEDBERR::eQueryInterface,
"CheckSPVersion()");
    }

    // Now fill the binding information
for result set 1 output columns

```

```

    InitBindings(&acOutputDBBinding[0
], nOutputParams, eOutputColumn);
    // Binding for a rowset
    SetBinding(&acOutputDBBinding[0],
0, sizeof(db_sp_version), DBTYPE_STR);

    hr = pIAccessor->CreateAccessor(
DBACCESSOR_ROWDATA,
nOutputParams,
acOutputDBBinding,
sizeof(db_sp_version),
&hTpccVersionOutputAccessor,
acOutputDBBindStatus);
    if (FAILED(hr))
    {
        ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"CheckSPVersion()");
    }

    hr = pICommandText-
>Execute(NULL, IID_IRowset, NULL, NULL,
(IUnknown **)&pRowset);
    if (FAILED(hr))
    {
        ThrowError(pICommandText,
COLEDBERR::eExecute, "CheckSPVersion()");
    }

    // Fetch the result row handle(s)
    hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows,
&cRowsObtained, &prghRow);
    if (FAILED(hr))
    {
        ThrowError(pICommandText,
COLEDBERR::eGetNextRows,
"CheckSPVersion()");
    }

    // Fetch the actual row data by
handle
    hr = pRowset->GetData(rghRow,
hTpccVersionOutputAccessor, &db_sp_version);
    if (FAILED(hr))
    {
        ThrowError(pICommandText,
COLEDBERR::eGetData, "CheckSPVersion()");
    }

    // Release row(s)
    hr = pRowset->Release();

    pICommandText->Release();

    // Check the retrieved version
    if (strcmp(db_sp_version, sVersion))
        throw

    new CTPCC_OLEDB_ERR(
CTPCC_OLEDB_ERR::ERR_WRONG_SP_VERSION
);
}

void CTPCC_OLEDB::ThrowError( IUnknown*
pObjectWithError, COLEDBERR::ACTION eAction,
LPCTSTR szLocation)
{

```

```

    HRESULT
    //char
    hr;

    szState[6];
    char

    szMsg[SQL_MAX_MESSAGE_LEN*
H];
    char

    szTmp[6*SQL_MAX_MESSAGE_LEN
GTH];

    COLEDBERR
    *pOLEDBErr;
    // not allocated until
needed (maybe never)
    int

    iLen;
    // Interfaces
    IErrorInfo*
    pIErrorInfoAll
= NULL;
    IErrorInfo*
    pIErrorInfoRecord = NULL;
    IErrorRecords*
    pIErrorRecords = NULL;
    ISupportErrorInfo*
    pISupportErrorInfo = NULL;
    ISQLServerErrorInfo*
    pISQLServerErrorInfo = NULL;
    ISQLErrorInfo*
    pISQLErrorInfo =
NULL;

    // Information used when cannot
get custom error object
    ERRORINFO
    {
        BasicErrorInfo;
        BSTR

        bstrDescription;
        // Number of error records.
        ULONG
        nRecs;
        ULONG
        nRec;

        // SQL Server error information
from ISQLServerErrorInfo.
        SSERERRORINFO*
        pSSerErrorInfo = NULL;
        OLECHAR*
        pSSerErrorStrings = NULL;

        assert(pObjectWithError != NULL);

        pOLEDBErr = new
COLEDBERR(szLocation);

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

        szTmp[0] = 0;

        // Only ask for error information if
the interface supports it.
        // Note: SQLOLEDB provider
supports error interface, so this check is
// for good style only.
        hr = pObjectWithError-
>QueryInterface(IID_ISupportErrorInfo, (void**)
&pISupportErrorInfo);
        if (FAILED(hr))

```

```

    {
        _snprintf(szMsg,
sizeof(szMsg), "SupportErrorInfo interface not
supported (hr=0x%X)", hr);
        pOLEDBErr-
>m_OLEDBErrStr = new char[strlen(szMsg)+1];
        strcpy(pOLEDBErr-
>m_OLEDBErrStr, szMsg);
        throw pOLEDBErr;
    }
    /*if (FAILED(pISupportErrorInfo-
>InterfaceSupportsErrorInfo(IID_InterfaceWithEr
ror)))
    {
        _snprintf(szMsg,
sizeof(szMsg), "InterfaceWithError interface not
supported");
        pOLEDBErr-
>m_OLEDBErrStr = new char[strlen(szMsg)+1];
        strcpy(pOLEDBErr-
>m_OLEDBErrStr, szMsg);
        return;
    }*/

    // Do not test the return of
GetErrorInfo. It can succeed and return
// a NULL pointer in pIErrorInfoAll.
Simply test the pointer.
GetErrorInfo(0, &pIErrorInfoAll);

    if (pIErrorInfoAll != NULL)
    {
        // Test to see if it's a
valid OLE DB IErrorInfo interface
// exposing a list of
records.
        if
(SUCCEEDED(pIErrorInfoAll-
>QueryInterface(IID_IErrorRecords, (void**)
&pIErrorRecords)))
        {
            pIErrorRecords-
>GetRecordCount(&nRecs);

            // Within
each record, retrieve information from each
// of the
defined interfaces.
            for
(nRec = 0; nRec < nRecs; nRec++)
            {
                // Request the generic SQL error
interface.
                pIErrorRecords-
>GetCustomErrorObject(nRec,

                IID_ISQLErrorInfo, //
generic SQL error interface

                (IUnknown**)
&pISQLErrorInfo);

                if (pISQLErrorInfo != NULL)
                {

                    // Request SQL
Server-specific error interface, not the generic
SQL error interface.

```

```

        pISQLErrorInfo-
>QueryInterface(
        IID_ISQLErrorInfo, // SQL
Server error interface

        (void**)
&pISQLErrorInfo);
    }

    // Test to ensure the reference is
valid, then

    // get error information from
ISQLErrorInfo.
    if (pISQLErrorInfo != NULL)
    {
        pISQLErrorInfo-
>GetErrorInfo(&pSSErrorInfo, &pSSErrorStrings);

        //
ISQLErrorInfo::GetErrorInfo succeeds

        // even when it has
nothing to return. Test the

        // pointers before
using.
        if (pSSErrorInfo)
        {
            // First,
add the error message.

            //
Convert Unicode error string to ANSI.
            WideCharToMultiByte(CP_THREAD_
ACP, 0,

            pSSErrorInfo->pwszMessage, -1,

            szMsg, sizeof(szMsg),

            NULL, NULL);

            // 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, "\r\n");

        //
concatenate the error record to the overall error
message

        strcat(
szTmp, szMsg );

        //
Second, add the stored procedure name and line
number, if available.

        if
(wcslen(pSSErrorInfo->pwszProcedure)>0)
        {

            // Prefix with a line break

            iLen = sprintf(szMsg,
"\r\nProcedure: ");

            // Convert Unicode error string to
ANSI.
            WideCharToMultiByte(CP_THREAD_
ACP, 0,

            pSSErrorInfo-
>pwszProcedure, -1,

            &szMsg[iLen],
sizeof(szMsg) - iLen,

            NULL, NULL);

            // Check if have space to add the
line number.

            // Assume the line number takes no
more than 3 digits.

            if ((strlen(szMsg) + 4)<
sizeof(szMsg))
            {

```

```

        _snprintf(&szMsg[strlen(szMsg)],
sizeof(szMsg),
        "%d",
pSSErrorInfo->wLineNumber);
    }

    // quit if there isn't enough room to
concatenate error text

    if ( (strlen(szMsg) + 2) >
(sizeof(szTmp) - strlen(szTmp)))

        break;

    // concatenate the error record to
the overall error message

    strcat( szTmp, szMsg );

    // copy the overall error string to
the exception

    pOLEDBErr->m_OLEDBErrStr =
new char[strlen(szTmp)+1];

    strcpy(pOLEDBErr-
>m_OLEDBErrStr, szTmp);
}

// Third,
capture the (first) database error

if
(pOLEDBErr->m_NativeError == 0 &&
pSSErrorInfo->INative != 0)
{

    pOLEDBErr->m_NativeError =
pSSErrorInfo->INative;

    // Check for deadlock error code
and set the deadlock flag

    if (pSSErrorInfo->INative == 1205)

    {

```

```

        pOLEDBErr-
>m_bDeadLock = TRUE;
    }
}

//
IMalloc::Free needed to release references
// on
returned values.

if
(m_pIMalloc != NULL)
{

    m_pIMalloc-
>Free(pSSErrorStrings);

    m_pIMalloc->Free(pSSErrorInfo);
}

}

pISQLServerErrorInfo-
>Release();
}
else
{
    // Custom error object
is not supported.

    // Use general OLE-
DB error interface.

    // Get the numeric
error code

    pErrorRecords-
>GetBasicErrorInfo(nRec, &BasicErrorInfo);

    if (pOLEDBErr-
>m_NativeError == 0)
    {
        // Get
the failed call HRESULT code, which is not really
the native error

        pOLEDBErr->m_NativeError =
BasicErrorInfo.hrError;
    }
}

```

```

        // Try to get the string
description of the error.

        pErrorRecords-
>GetErrorInfo(nRec, LOCALE_USER_DEFAULT,
(IErrorInfo**) &pIErrorInfoRecord);

        if (pIErrorInfoRecord)
        {

            pIErrorInfoRecord-
>GetDescription(&bstrDescription);

            //
Convert Unicode error string to ANSI.

            WideCharToMultiByte(CP_THREAD_
ACP, 0,
            bstrDescription, -1,
            szMsg, sizeof(szMsg),
            NULL, NULL);

            pOLEDBErr->m_OLEDBErrStr =
new char[strlen(szMsg)+1];

            strcpy(pOLEDBErr-
>m_OLEDBErrStr, szMsg);
        }
    }
} // if
(SUCCEEDED(pIErrorInfoAll-
>QueryInterface(IID_IErrorRecords, (void**)
&pIErrorRecords))
else
{ // No
IErrorRecords interface supported. Use default
IErrorInfo.

    // Note:
SQLOLEDB supports IErrorRecords, so this check
is for good style only.

    _snprintf(szMsg, sizeof(szMsg),
"IErrorRecords interface not supported");

    pOLEDBErr->m_OLEDBErrStr =
new char[strlen(szMsg)+1];

    strcpy(pOLEDBErr-
>m_OLEDBErrStr, szMsg);
}
}

```



```

        pIErrorInfoAll-
>Release();
    } // if (pIErrorInfoAll !=
NULL)
    else
    {
        // No IErrorInfo
interface supported.
        // Note: SQLOLEDB
supports IErrorInfo, so this check is for good
style only.
        _snprintf(szMsg,
sizeof(szMsg), "IErrorInfo interface not
supported");
        pOLEDBErr-
>m_OLEDBErrStr = new char[strlen(szMsg)+1];
        strcpy(pOLEDBErr-
>m_OLEDBErrStr, szMsg);
    }
    throw pOLEDBErr;
}
/*
*
* Create a new command object from
the SQL text passed in.
*
*/
void CTPCC_OLEDB::CreateCommand(wchar_t*
szSQLCommand,
    // I: SQL query for the command

    ICommandText**
ppICommandText // O: returned
command object

)
{
    HRESULT
    hr;

    // Create a new command object
hr = m_pIDBCreateCommand-
>CreateCommand(NULL, IID_ICommandText,
(IUnknown**)ppICommandText);
    if (FAILED(hr))
    {
        ThrowError(m_pIDBCreateComman-
d, COLEDBERR::eCreateCommand,
"CTPCC_OLEDB::CreateCommand");
    }

    // Set command text
hr = (*ppICommandText)-
>SetCommandText(DBGUID_SQL,
szSQLCommand);
    if (FAILED(hr))
    {
        ThrowError(*ppICommandText,
COLEDBERR::eSetCommandText,
"CTPCC_OLEDB::CreateCommand");
    }

    // Prepare the command
PrepareCommand(*ppICommandTe-
xt);
}
/*

```

```

* QueryInterface and Prepare in one
function for SQLSERVER PREPARE property is set
to off to prepare immediately.
*/
void
CTPCC_OLEDB::PrepareCommand(ICommandTex-
t* pICommandText)
{
    HRESULT
    hr;
    ICommandPrepare*
pICommandPrepare;
    ICommandProperties*
pICommandProperties;
    DBPROPSET
rowSetPropSet;
    DBPROP
rowSetProp;

    // Set the deferred prepare
property to false.
rowSetProp.dwPropertyID =
SSPROP_DEFERPREPARE;

    memset(&rowSetProp.vValue, 0,
sizeof(rowSetProp.vValue));

    rowSetProp.dwOptions =
DBPROPOPTIONS_REQUIRED;
    rowSetProp.colid = DB_NULLID;

    rowSetPropSet.cProperties = 1;
    rowSetPropSet.guidPropertySet =
DBPROPSET_SQLSERVERROWSET;
    rowSetPropSet.rgProperties =
&rowSetProp;

    // Query interface for setting
properties
hr = pICommandText-
>QueryInterface(IID_ICommandProperties, (void
**)&pICommandProperties);
    if (FAILED(hr))
    {
        ThrowError(pICommandText,
COLEDBERR::eQueryInterface,
"CTPCC_OLEDB::PrepareCommand");
    }

    // Set the property set
hr = pICommandProperties-
>SetProperties(1, &rowSetPropSet);
    if (FAILED(hr))
    {
        ThrowError(pICommandText,
COLEDBERR::eQueryInterface,
"CTPCC_OLEDB::PrepareCommand");
    }

    // Get interface for preparing
commands
hr = pICommandText-
>QueryInterface(IID_ICommandPrepare, (void
**)&pICommandPrepare);
    if (FAILED(hr))
    {
        ThrowError(pICommandText,
COLEDBERR::eQueryInterface,
"CTPCC_OLEDB::PrepareCommand");
    }

    // Prepare Payment command

```

```

hr = pICommandPrepare-
>Prepare(0, hr);
}

ThrowError(pICommandPrepare,
COLEDBERR::ePrepare,
"CTPCC_OLEDB::PrepareCommand");
}

/*
* Initialize fields of an array of
bindings structures.
* Needs to be called before setting
individual parameter/column bindings.
*/
void CTPCC_OLEDB::InitBindings(DBBINDING*
pDBBindings, // IO: array of bindings

    int iCount, // I:
number of elements in the array

    eBindingType BindingType)
    // I: what the bindings will be used
for (parameters/columns)
{
    int i;

    for(i = 0; i < iCount; i++)
    {
        pDBBindings[i].iOrdinal = i + 1;
        pDBBindings[i].obLength = 0;
        pDBBindings[i].obStatus = 0;
        pDBBindings[i].pTypeInfo = NULL;
        pDBBindings[i].pObject = NULL;
        pDBBindings[i].pBindExt = NULL;
        pDBBindings[i].dwPart = DBPART_VALUE;

        switch (BindingType)
        {
            case
eInputParameter:
                pDBBindings[i].eParamIO =
DBPARAMIO_INPUT;
                break;

            case
eOutputParameter:
                pDBBindings[i].eParamIO =
DBPARAMIO_OUTPUT;
                break;

            case
eInputOutputParameter:
                pDBBindings[i].eParamIO =
DBPARAMIO_INPUT | DBPARAMIO_OUTPUT;
                break;
                case eOutputColumn:
                    pDBBindings[i].eParamIO =
DBPARAMIO_NOTPARAM;
                    break;

            default:
                assert(false); // this should never
happen
        }
    }
}

```

```

    pDBBindings[i].dwMemOwner =
DBMEMOWNER_CLIENTOWNED;
    pDBBindings[i].dwFlags = 0;

    pDBBindings[i].bPrecision = 0;
    pDBBindings[i].bScale = 0;
}
}

/*
 * Perform binding for one parameter
or output column.
 *
 */
void CTPCC_OLEDB::SetBinding(DBBINDING*
pDBBinding, // I: binding row structure

    size_t obValue,
    // I: parameter
(column) offset in the user buffer

    size_t cbMaxLen,
    // I: parameter (column) length

    DBTYPE wType
    // I: parameter
(column) type

)
{
    pDBBinding->obValue =
(ULONG)obValue;
    pDBBinding->cbMaxLen =
(ULONG)cbMaxLen;
    pDBBinding->wType = wType;
}

void CTPCC_OLEDB::InitStockLevelParams()
{
    int
    i;
    HRESULT
    hr;
    wchar_t
    szName[iMAX_SP_NAME_LEN];
    IAccessor*
    piAccessor;

    const ULONG
    nInputParams = 3;
    // input parameters
    const ULONG
    nOutputParams = 1;
    // output 1st result set columns

    // Structure to bind in accessor
    DBBINDING
    acInputDBBinding[nInputParams];
    DBBINDSTATUS
    acInputDBBindStatus[nInputParams];
    DBBINDING
    acOutputDBBinding[nOutputParams];
};

```

```

    DBBINDSTATUS
    acOutputDBBindStatus[nOutputPara
ms];
    // Set command text
    _snwprintf(szName,
sizeof(szName)/sizeof(szName[0]),
    L"(call
%stpcck_stocklevel (?,?,?))", m_szSPPrefix);

    // Create and Prepare a new
command object for StockLevel.
    CreateCommand(szName,
&m_piStockLevelCommand);

    // Describe the consumer buffer by
filling in the array
    // of DBBINDING structures. Each binding
associates
    // a single parameter to the consumer's buffer.
    InitBindings(&acInputDBBinding[0],
nInputParams, eInputParameter);

    i = 0;
    // StockLevel parameter 1
    SetBinding(&acInputDBBinding[i++
], offsetof(STOCK_LEVEL_DATA, w_id),
sizeof(m_bxn.StockLevel.w_id), DBTYPE_I4);

    // StockLevel parameter 2
    SetBinding(&acInputDBBinding[i++
], offsetof(STOCK_LEVEL_DATA, d_id),
sizeof(m_bxn.StockLevel.d_id), DBTYPE_UI1);

    // StockLevel parameter 3
    SetBinding(&acInputDBBinding[i++
], offsetof(STOCK_LEVEL_DATA, threshold),
sizeof(m_bxn.StockLevel.threshold), DBTYPE_I2);

    hr = m_piStockLevelCommand-
>QueryInterface(IID_IAccessor, (void
**)&piAccessor);
    if (FAILED(hr))
    {
        ThrowError(m_piStockLevelComma
nd, COLEDBERR::eQueryInterface,
"InitStockLevelParams()");
    }

    hr = piAccessor->CreateAccessor(
DBACCESSOR_PARAMETERDATA,
nInputParams,
acInputDBBinding,
sizeof(STOCK_LEVEL_DATA),
&m_hStockLevelInputAccessor,
acInputDBBindStatus);
    if (FAILED(hr))
    {
        ThrowError(piAccessor,
COLEDBERR::eCreateAccessor,
"InitStockLevelParams()");
    }

    m_StockLevelExecuteParams.cPara
mSets = 1;
    m_StockLevelExecuteParams.hAcce
ssor = m_hStockLevelInputAccessor;

```

```

    m_StockLevelExecuteParams.pData
= &m_bxn.StockLevel;
    // Now fill the binding information
for result set 1 output columns
    InitBindings(&acOutputDBBinding[0
], nOutputParams, eOutputColumn);

    // Binding for a rowset that may
return more than one row.
    i = 0;
    // StockLevel output column 1
    SetBinding(&acOutputDBBinding[i+
+], offsetof(STOCK_LEVEL_DATA, low_stock),
sizeof(m_bxn.StockLevel.low_stock),
DBTYPE_I4);

    hr = piAccessor->CreateAccessor(
DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
nOutputParams,
acOutputDBBinding,
sizeof(STOCK_LEVEL_DATA),
&m_hStockLevelOutputAccessor,
acOutputDBBindStatus);
    if (FAILED(hr))
    {
        ThrowError(piAccessor,
COLEDBERR::eCreateAccessor,
"InitStockLevelParams()");
    }
}

void CTPCC_OLEDB::StockLevel()
{
    HRESULT
    hr;
    int
    iTryCount = 0;

    IRowset*
    pRowset;
    LONG
    cRows = 1; // number of rows
returned in the rowset
    ULONG
    cRowsObtained;
    HROW
    rghRow;
    //returned row handles
    HROW*
    prghRow = &rghRow;

    while (TRUE)
    {
        try
        {
            //
            Execute the prepared command
            hr =
m_piStockLevelCommand->Execute(NULL,
IID_IRowset, &m_StockLevelExecuteParams,
NULL,

            (IUnknown
**)&pRowset);
            if
(FAILED(hr))
            {

```

```

        ThrowError(m_pIStockLevelComma
nd, COLEDBERR::eExecute, "StockLevel()");
    }
    // Fetch
the result row handle(s)
    hr =
pRowset->GetNextRows(DB_NULL_HCHAPTER,
0, cRows, &cRowsObtained, &prghRow);
    if
(FAILED(hr))
    {
        ThrowError(m_pIStockLevelComma
nd, COLEDBERR::eGetNextRows, "StockLevel()");
    }
    // Fetch
the actual row data by handle
    hr =
pRowset->GetData(rghRow,
m_hStockLevelOutputAccessor,
&m_txn.StockLevel);
    if
(FAILED(hr))
    {
        ThrowError(m_pIStockLevelComma
nd, COLEDBERR::eGetData, "StockLevel()");
    }
    //
Release row(s)
    hr =
pRowset->ReleaseRows(cRowsObtained,
prghRow, NULL, NULL, NULL);
    //
Release rowset
    hr =
pRowset->Release();

    m_txn.StockLevel.exec_status_code
= eOK;
    break;
    }
    catch (COLEDBERR
*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_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_R
ETRIED_TRANS, iTryCount);
}

void CTPCC_OLEDB::InitNewOrderParams()
{

```

```

    int
nResultCnt;
    wchar_t
szName[IMAX_SP_NAME_LEN];
    IAccessor*
pIAccessor;

    const ULONG
nInputParams = 5 +
3*MAX_OL_NEW_ORDER_ITEMS; // input
parameters

    const ULONG
nOutputParams = 5;
    // output 1st result set columns
    const ULONG
nOutputParams2 = 8;
    // output 2nd result set columns
    // Structure to bind in accessor
    DBBINDING
acInputDBBinding[nInputParams];
    DBBINDSTATUS
acInputDBBindStatus[nInputParams
];
    DBBINDING
acOutputDBBinding[nOutputParams
];
    DBBINDSTATUS
acOutputDBBindStatus[nOutputPara
ms];
    DBBINDING
acOutputDBBinding2[nOutputParam
s2];
    DBBINDSTATUS
acOutputDBBindStatus2[nOutputPar
ams2];

    // Describe the consumer buffer by
filling in the array
    // of DBBINDING structures. Each binding
associates
    // a single parameter to the consumer's buffer.
    InitBindings(&acInputDBBinding[0],
nInputParams, eInputParameter);

    i = 0;
    // NewOrder parameter 1
    SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA, w_id),
sizeof(m_txn.NewOrder.w_id), DBTYPE_I4);

    // NewOrder parameter 2
    SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA, d_id),
sizeof(m_txn.NewOrder.d_id), DBTYPE_UI1);

    // NewOrder parameter 3
    SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA, c_id),
sizeof(m_txn.NewOrder.c_id), DBTYPE_I4);

    // NewOrder parameter 4
    SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA, o_ol_cnt),
sizeof(m_txn.NewOrder.o_ol_cnt), DBTYPE_UI1);

```

```

    // NewOrder parameter 5
    SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA, o_all_local),
sizeof(m_txn.NewOrder.o_all_local),
DBTYPE_UI1);

    for (j=0;
j<MAX_OL_NEW_ORDER_ITEMS; j++)
    {
        SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA, OL[j].ol_i_id),
sizeof(m_txn.NewOrder.OL[j].ol_i_id),
DBTYPE_I4);

        SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA,
OL[j].ol_supply_w_id),
sizeof(m_txn.NewOrder.OL[j].ol_supply_w_id),
DBTYPE_I4);

        SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA,
OL[j].ol_quantity),
sizeof(m_txn.NewOrder.OL[j].ol_quantity),
DBTYPE_I2);
    }

    // Now fill the binding information
for result set 1 output columns
    InitBindings(&acOutputDBBinding[0
], nOutputParams, eOutputColumn);

    // Binding for the order line rowsets
(each consist of one row).
    // Bind to offsets of the
OL_NEW_ORDER_DATA structure instead of
NEW_ORDER_DATA.
    // IRowset::GetData() will be
passed individual array slots OL[i] to fetch the
data
    // from the row set.

    i = 0;
    // NewOrder output column 1
    SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_NEW_ORDER_DATA, ol_i_name),
sizeof(m_txn.NewOrder.OL[0].ol_i_name),
DBTYPE_STR);

    // NewOrder output column 2
    SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_NEW_ORDER_DATA, ol_stock),
sizeof(m_txn.NewOrder.OL[0].ol_stock),
DBTYPE_I2);

    // NewOrder output column 3
    SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_NEW_ORDER_DATA,
ol_brand_generic),
sizeof(m_txn.NewOrder.OL[0].ol_brand_generic),
DBTYPE_STR);

    // NewOrder output column 4
    SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_NEW_ORDER_DATA, ol_i_price),
sizeof(m_txn.NewOrder.OL[0].ol_i_price),
DBTYPE_R8);

    // NewOrder output column 5

```

```

        SetBinding(&acOutputDBBinding[i+
++], offsetof(OL_NEW_ORDER_DATA,
ol_amount),
sizeof(m_bxn.NewOrder.OL[0].ol_amount),
DBTYPE_R8);

// Now fill the binding information
for result set 2 output columns
InitBindings(&acOutputDBBinding2[
0], nOutputParams2, eOutputColumn);

i = 0;
// NewOrder output column 1
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA, w_tax),
sizeof(m_bxn.NewOrder.w_tax), DBTYPE_R8);

// NewOrder output column 2
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA, d_tax),
sizeof(m_bxn.NewOrder.d_tax), DBTYPE_R8);

// NewOrder output column 3
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA, o_id),
sizeof(m_bxn.NewOrder.o_id), DBTYPE_I4);

// NewOrder output column 4
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA, c_last),
sizeof(m_bxn.NewOrder.c_last), DBTYPE_STR);

// NewOrder output column 5
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA, c_discount),
sizeof(m_bxn.NewOrder.c_discount),
DBTYPE_R8);

// NewOrder output column 6
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA, c_credit),
sizeof(m_bxn.NewOrder.c_credit), DBTYPE_STR);

// NewOrder output column 7
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA, o_entry_d),
sizeof(m_bxn.NewOrder.o_entry_d),
DBTYPE_DBTIMESTAMP);

// NewOrder output column 8
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA,
o_commit_flag),
sizeof(m_bxn.NewOrder.o_commit_flag),
DBTYPE_I2);

for (j=0;
j<MAX_OL_NEW_ORDER_ITEMS; j++)
{
// Set command text
first

// Print the fixed first
portion of parameters
i =
_snowprintf(szName,
sizeof(szName)/sizeof(szName[0]),

L"{call
%stppcc_neworder (?,?,?,?,, m_szSPPrefix);

```

```

// Now print the
variable portion depending on the number of
order line parameters for (iOLCount = 0;
iOLCount <= j; ++iOLCount)
{
i +=
_snowprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L",?,?,?");
}

// Print the fixed end
if (j !=
MAX_OL_NEW_ORDER_ITEMS - 1)
{
//
append 'default' for the parameters that are not
used
i +=
_snowprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i,
L",default");
}
else // using
all 15 order line parameters
{
i +=
_snowprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L"");
}

// Create and Prepare
a new command object for NewOrder.
CreateCommand(szName,
&m_piNewOrderCommand[j]);

// Now create the
input accessor for this prepared command
hr =
m_piNewOrderCommand[j]-
>QueryInterface(IID_IAccessor, (void
**)&piAccessor);

if (FAILED(hr))
{
ThrowError(m_piNewOrderComma
nd[j], COLEDBERR::eQueryInterface,
"InitNewOrderParams()");
}

hr = piAccessor-
>CreateAccessor(

DBACCESSOR_PARAMETERDATA,

5 + 3 * (j + 1),

acInputDBBinding,

sizeof(NEW_ORDER_DATA),

```

```

&m_hNewOrderInputAccessor[j],

acInputDBBindStatus);
if (FAILED(hr))
{
ThrowError(piAccessor,
COLEDBERR::eCreateAccessor,
"InitNewOrderParams()");
}

m_NewOrderExecuteParams[j].cPar
amSets = 1;
//
m_NewOrderExecuteParams.hAccessor is set
dynamically at run-time
// based on the
number of new order items for the particular
transaction call.

m_NewOrderExecuteParams[j].hAcc
essor = m_hNewOrderInputAccessor[j];

m_NewOrderExecuteParams[j].pDat
a = &m_bxn.NewOrder;

// Create accessor for
the first rowset
hr = piAccessor-
>CreateAccessor(

DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,

nOutputParams,

acOutputDBBinding,

sizeof(OL_NEW_ORDER_DATA),

&m_hNewOrderOutputAccessor[j],

acOutputDBBindStatus);
if (FAILED(hr))
{
ThrowError(piAccessor,
COLEDBERR::eCreateAccessor,
"InitNewOrderParams()");
}

// Create accessor for
the second rowset
hr = piAccessor-
>CreateAccessor(

DBACCESSOR_ROWDATA, //
cannot be optimized too because #1 accessor is

nOutputParams2,

acOutputDBBinding2,

sizeof(NEW_ORDER_DATA),

&m_hNewOrderOutputAccessor2[j],

```

```

        acOutputDBBindStatus2);
        if (FAILED(hr))
        {
            ThrowError(pIAccessor,
                COLEDBERR::eCreateAccessor,
                "InitNewOrderParams()");
        }

        pIAccessor->Release();
    }

void CTPCC_OLEDB::NewOrder()
{
    HRESULT hr;
    int cRows = 1; // number of rows returned in the 1st rowset
    ULONG cRowsObtained;
    HROW rghRows; //returned row handles for the 1st result set
    HROW* prghRows = &rghRows;
    LONG cRows2 = 1; // number of rows returned in the 2nd rowset
    ULONG cRowsObtained2;
    HROW rghRows2; //returned row handle for the 2nd result set
    HROW* prghRows2 = &rghRows2;
    int i;
    long lRowsAffected; // the number of affected rows for a rowset
    int iHandleIndex; // index into the handle arrays based on the orders count

    // 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;
        }
    }

    iHandleIndex = m_txn.NewOrder.o_ol_cnt - 1; // for convenience

    while (TRUE)
    {
        try
        {
            // Execute the prepared command (according to the number of new orders)
            // Ask for IMultipleResults because it returns 2 rowsets.
            hr = m_pINewOrderCommand[iHandleIndex]->Execute(
                NULL,
                IID_IMultipleResults,
                &m_NewOrderExecuteParams[iHandleIndex],
                NULL,
                (IUnknown**) &pMultipleResults);
            if (FAILED(hr))
            {
                ThrowError(m_pINewOrderCommand[iHandleIndex], COLEDBERR::eExecute, "NewOrder()");
            }

            // Get order line results
            // Get the first rowset object
            hr = pMultipleResults->GetResult(NULL, 0, IID_IRowset, &lRowsAffected, (IUnknown**) &pRowset);
            if (FAILED(hr))
            {

```

```

                char szTmp[256];
                _snprintf(szTmp, sizeof(szTmp), "NewOrder() result set %d, hr=0x%X", i, hr);

                ThrowError(m_pINewOrderCommand[m_txn.NewOrder.o_ol_cnt - 1], COLEDBERR::eGetResult, szTmp);
            }

            // Fetch the result row handle(s)
            hr = pRowset->GetNextRows(DB_NULL_HCHAPTER, 0, cRows, &cRowsObtained, &prghRows);
            if (FAILED(hr))
            {
                ThrowError(m_pINewOrderCommand[iHandleIndex], COLEDBERR::eGetNextRows, "NewOrder()");
            }

            // Fetch the actual row data by handle
            hr = pRowset->GetData(rghRows, m_hNewOrderOutputAccessor[iHandleIndex], &m_txn.NewOrder.OL[i]);
            if (FAILED(hr))
            {
                ThrowError(m_pINewOrderCommand[iHandleIndex], COLEDBERR::eGetData, "NewOrder()");
            }

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

            // Release row(s)
            hr = pRowset->ReleaseRows(cRowsObtained, prghRows, NULL, NULL, NULL);
            // Release rowset
            hr = pRowset->Release();
        }

        // Get the second rowset object

```



```

        ThrowError(pIAccessor,
        COLEDBERR::eCreateAccessor,
        "InitPaymentParams()");
    }

    m_PaymentExecuteParams.cParam
Sets = 1;
    m_PaymentExecuteParams.hAccess
or = m_hPaymentInputAccessor;
    m_PaymentExecuteParams.pData =
&m_bxn.Payment;

    // Now fill the binding information
for output columns
    InitBindings(&acOutputDBBinding[0
], nOutputParams, eOutputColumn);

    i = 0;
    // Payment output column 1

    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_id),
sizeof(m_bxn.Payment.c_id), DBTYPE_I4);

    // Payment output column 2
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_last),
sizeof(m_bxn.Payment.c_last), DBTYPE_STR);

    // Payment output column 3
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, h_date),
sizeof(m_bxn.Payment.h_date),
DBTYPE_DBTIMESTAMP);

    // Payment output column 4
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, w_street_1),
sizeof(m_bxn.Payment.w_street_1),
DBTYPE_STR);

    // Payment output column 5
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, w_street_2),
sizeof(m_bxn.Payment.w_street_2),
DBTYPE_STR);

    // Payment output column 6
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, w_city),
sizeof(m_bxn.Payment.w_city), DBTYPE_STR);

    // Payment output column 7
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, w_state),
sizeof(m_bxn.Payment.w_state), DBTYPE_STR);

    // Payment output column 8
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, w_zip),
sizeof(m_bxn.Payment.w_zip), DBTYPE_STR);

    // Payment output column 9
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_street_1),
sizeof(m_bxn.Payment.d_street_1),
DBTYPE_STR);

    // Payment output column 10
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_street_2),
sizeof(m_bxn.Payment.d_street_2),
DBTYPE_STR);

```

```

    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_city),
sizeof(m_bxn.Payment.d_city), DBTYPE_STR);

    // Payment output column 12
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_state),
sizeof(m_bxn.Payment.d_state), DBTYPE_STR);

    // Payment output column 13
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_zip),
sizeof(m_bxn.Payment.d_zip), DBTYPE_STR);

    // Payment output column 14
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_first),
sizeof(m_bxn.Payment.c_first), DBTYPE_STR);

    // Payment output column 15
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_middle),
sizeof(m_bxn.Payment.c_middle), DBTYPE_STR);

    // Payment output column 16
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_street_1),
sizeof(m_bxn.Payment.d_street_1),
DBTYPE_STR);

    // Payment output column 17
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_street_2),
sizeof(m_bxn.Payment.d_street_2),
DBTYPE_STR);

    // Payment output column 18
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_city),
sizeof(m_bxn.Payment.d_city), DBTYPE_STR);

    // Payment output column 19
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_state),
sizeof(m_bxn.Payment.d_state), DBTYPE_STR);

    // Payment output column 20
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_zip),
sizeof(m_bxn.Payment.d_zip), DBTYPE_STR);

    // Payment output column 21
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_phone),
sizeof(m_bxn.Payment.c_phone), DBTYPE_STR);

    // Payment output column 22
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_since),
sizeof(m_bxn.Payment.c_since),
DBTYPE_DBTIMESTAMP);

    // Payment output column 23
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_credit),
sizeof(m_bxn.Payment.c_credit), DBTYPE_STR);

    // Payment output column 24
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_credit_lim),
sizeof(m_bxn.Payment.c_credit_lim),
DBTYPE_R8);

```

```

    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_discount),
sizeof(m_bxn.Payment.c_discount), DBTYPE_R8);

    // Payment output column 26
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_balance),
sizeof(m_bxn.Payment.c_balance), DBTYPE_R8);

    // Payment output column 27
    SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_data),
sizeof(m_bxn.Payment.c_data), DBTYPE_STR);

    hr = pIAccessor->CreateAccessor(
        DBACCESSOR_ROWDATA |
        DBACCESSOR_OPTIMIZED,
        nOutputParams,
        acOutputDBBinding,
        sizeof(PAYMENT_DATA),

&m_hPaymentOutputAccessor,
        acOutputDBBindStatus);
    if (FAILED(hr))
    {
        ThrowError(pIAccessor,
        COLEDBERR::eCreateAccessor,
        "InitPaymentParams()");
    }
}

void CTPCC_OLEDB::Payment()
{
    HRESULT
hr;
    int
    iTryCount = 0;

    IRowset*
pRowset;
    LONG
cRows = 1; // number of rows
returned in the rowset
    ULONG
cRowsObtained;
    HROW
rghRow;
    //returned row handles
    HROW*
prghRow = &rghRow;

    if (m_bxn.Payment.c_id != 0)
        m_bxn.Payment.c_last[0] = 0;

    while (TRUE)
    {
        try
        {
            //
            Execute the prepared command

            hr =
m_pIPaymentCommand->Execute(NULL,
IID_IRowset, &m_PaymentExecuteParams,
NULL,

            (IUnknown
**)&pRowset);

            if
(FAILED(hr))

```

```

    {
        ThrowError(m_pIPaymentCommand
, COLEDBERR::eExecute, "Payment()");
    }

    // Fetch
the result row handle(s)
    hr =
pRowset->GetNextRows(DB_NULL_HCHAPTER,
0, cRows, &cRowsObtained, &prghRow);
    if
(FAILED(hr))
    {
        ThrowError(m_pIPaymentCommand
, COLEDBERR::eGetNextRows, "Payment()");
    }

    // Fetch
the actual row data by handle
    hr =
pRowset->GetData(rghRow,
m_hPaymentOutputAccessor, &m_bxn.Payment);
    if
(FAILED(hr))
    {
        ThrowError(m_pIPaymentCommand
, COLEDBERR::eGetData, "Payment()");
    }

    //
Release row(s)
    hr =
pRowset->ReleaseRows(cRowsObtained,
prghRow, NULL, NULL, NULL);
    //
Release rowset
    hr =
pRowset->Release();

    if
(m_bxn.Payment.c_id == 0)
    {
        throw new CTPCC_OLEDB_ERR(
CTPCC_OLEDB_ERR::ERR_INVALID_CUST)
        else
            m_bxn.Payment.exec_status_code
= eOK;
    }
    break;
}
catch (COLEDBERR
*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_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_R
ETRIED_TRANS, iTryCount);

```

```

}
void CTPCC_OLEDB::InitOrderStatusParams()
{
    int
i;
HRESULT
hr;
wchar_t
wchar_t;
szName[iMAX_SP_NAME_LEN];
IAccessor*
pIAccessor;

    const ULONG
nInputParams = 4;
// input parameters
const ULONG
nOutputParams = 5;
// output 1st result set columns
const ULONG
nOutputParams2 = 8;
// output 2nd result set columns
// Structure to bind in accessor
DBBINDING
acInputDBBinding[nInputParams];
DBBINDSTATUS
acInputDBBindStatus[nInputParams
];
DBBINDING
acOutputDBBinding[nOutputParams
];
DBBINDSTATUS
acOutputDBBindStatus[nOutputPara
ms];
DBBINDING
acOutputDBBinding2[nOutputParam
s2];
DBBINDSTATUS
acOutputDBBindStatus2[nOutputPar
ams2];

    // Set command text
_snwprintf(szName,
sizeof(szName)/sizeof(szName[0]),
L"{call
%stpc_orderstatus(?,?,?)", m_szSPPrefix);

    // Create and Prepare a new
command object for OrderStatus.
CreateCommand(szName,
&m_pIOrderStatusCommand);

    // Describe the consumer buffer by
filling in the array
// of DBBINDING structures. Each binding
associates
// a single parameter to the consumer's buffer.
InitBindings(&acInputDBBinding[0],
nInputParams, eInputParameter);

    i = 0;
// OrderStatus parameter 1
SetBinding(&acInputDBBinding[i++
], offsetof(ORDER_STATUS_DATA, w_id),
sizeof(m_bxn.OrderStatus.w_id), DBTYPE_I4);

```

```

// OrderStatus parameter 2
SetBinding(&acInputDBBinding[i++
], offsetof(ORDER_STATUS_DATA, d_id),
sizeof(m_bxn.OrderStatus.d_id), DBTYPE_UI1);

// OrderStatus parameter 3
SetBinding(&acInputDBBinding[i++
], offsetof(ORDER_STATUS_DATA, c_id),
sizeof(m_bxn.OrderStatus.c_id), DBTYPE_I4);

// OrderStatus parameter 4
SetBinding(&acInputDBBinding[i++
], offsetof(ORDER_STATUS_DATA, c_last),
sizeof(m_bxn.OrderStatus.c_last), DBTYPE_STR);

    hr = m_pIOrderStatusCommand-
>QueryInterface(IID_IAccessor, (void
**)&pIAccessor);
    if (FAILED(hr))
    {
        ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eQueryInterface,
"InitOrderStatusParams()");
    }

    hr = pIAccessor->CreateAccessor(
DBACCESSOR_PARAMETERDATA,
nInputParams,
acInputDBBinding,
sizeof(ORDER_STATUS_DATA),
&m_hOrderStatusInputAccessor,
acInputDBBindStatus);
    if (FAILED(hr))
    {
        ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitOrderStatusParams()");
    }

    m_OrderStatusExecuteParams.cPar
amSets = 1;
    m_OrderStatusExecuteParams.hAcc
essor = m_hOrderStatusInputAccessor;
    m_OrderStatusExecuteParams.pDat
a = &m_bxn.OrderStatus;

    // Now fill the binding information
for result set 1 output columns

    InitBindings(&acOutputDBBinding[0
], nOutputParams, eOutputColumn);

    // Binding for a rowset that may
return more than one row.
// Bind to offsets of the
OL_ORDER_STATUS_DATA structure instead of
ORDER_STATUS_DATA.
// IRowset::GetData() will be
passed individual array slots OL[i] to fetch the
data
// from the row set.

    i = 0;
// OrderStatus output column 1
SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_ORDER_STATUS_DATA,
ol_supply_w_id),
sizeof(m_bxn.OrderStatus.OL[0].ol_supply_w_id),
DBTYPE_I4);

```



```

// OrderStatus output column 2
SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_ORDER_STATUS_DATA, ol_i_id),
sizeof(m_bxn.OrderStatus.OL[0].ol_i_id),
DBTYPE_I4);

// OrderStatus output column 3
SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_ORDER_STATUS_DATA,
ol_quantity),
sizeof(m_bxn.OrderStatus.OL[0].ol_quantity),
DBTYPE_I2);

// OrderStatus output column 4
SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_ORDER_STATUS_DATA,
ol_amount),
sizeof(m_bxn.OrderStatus.OL[0].ol_amount),
DBTYPE_R8);

// OrderStatus output column 5
SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_ORDER_STATUS_DATA,
ol_delivery_d),
sizeof(m_bxn.OrderStatus.OL[0].ol_delivery_d),
DBTYPE_DBTIMESTAMP);

hr = pIAccessor->CreateAccessor(
DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
nOutputParams,
acOutputDBBinding,
sizeof(OL_ORDER_STATUS_DATA),
&m_hOrderStatusOutputAccessor,
acOutputDBBindStatus);
if (FAILED(hr))
{
ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitOrderStatusParams()");
}

// Now fill the binding information
for result set 2 output columns
InitBindings(&acOutputDBBinding2[
0], nOutputParams2, eOutputColumn);

i = 0;
// OrderStatus output column 1
SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA, c_id),
sizeof(m_bxn.OrderStatus.c_id), DBTYPE_I4);

// OrderStatus output column 2
SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA, c_last),
sizeof(m_bxn.OrderStatus.c_last), DBTYPE_STR);

// OrderStatus output column 3
SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA, c_first),
sizeof(m_bxn.OrderStatus.c_first), DBTYPE_STR);

// OrderStatus output column 4
SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA, c_middle),
sizeof(m_bxn.OrderStatus.c_middle),
DBTYPE_STR);

// OrderStatus output column 5

```

```

SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA,
o_entry_d),
sizeof(m_bxn.OrderStatus.o_entry_d),
DBTYPE_DBTIMESTAMP);
// OrderStatus output column 7
SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA,
o_carrier_id),
sizeof(m_bxn.OrderStatus.o_carrier_id),
DBTYPE_I2);

// OrderStatus output column 8
SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA,
c_balance), sizeof(m_bxn.OrderStatus.c_balance),
DBTYPE_R8);

// OrderStatus output column 9
SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA, o_id),
sizeof(m_bxn.OrderStatus.o_id), DBTYPE_I4);

hr = pIAccessor->CreateAccessor(
DBACCESSOR_ROWDATA, //
cannot be optimized too because #1 accessor is
nOutputParams2,
acOutputDBBinding2,
sizeof(NEW_ORDER_DATA),
&m_hOrderStatusOutputAccessor2,
acOutputDBBindStatus2);
if (FAILED(hr))
{
ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitOrderStatusParams()");
}

void CTPCC_OLEDB::OrderStatus()
{
HRESULT hr;

int
iTryCount = 0;
IMultipleResults*
pMultipleResults;
IRowset*
pRowset;
IRowset*
pRowset2;
LONG
cRows =
MAX_OL_ORDER_STATUS_ITEMS; //
number of rows returned in the 1st rowset

ULONG
cRowsObtained;

HROW
rghRows[MAX_OL_ORDER_STATUS
_ITEMS]; //returned row
handles for the 1st result set
HROW*
prghRows =
&rghRows[0];
LONG
cRows2 = 1; //
number of rows returned in the 2nd rowset

```

```

ULONG
cRowsObtained2;
HROW
rghRows2;
//returned row handle for the 2nd
result set
HROW*
prghRows2 =
&rghRows2;
int
;
long
;
IRowsAffected;
// the number of affected rows for
a rowset
if (m_bxn.OrderStatus.c_id != 0)
m_bxn.OrderStatus.c_last[0] = 0;
while (TRUE)
{
try
{
Execute the prepared command //
// Ask
for IMultipleResults because it returns 2 rowsets.
hr =
m_pIOrderStatusCommand->Execute(NULL,
IID_IMultipleResults,
&m_OrderStatusExecuteParams, NULL,
(IUnknown
**)&pMultipleResults);
if
(FAILED(hr))
{
ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eExecute, "OrderStatus()");
}

////////////////////////////////////
// Get
order line results
////////////////////////////////////
// Get
the first rowset object
hr =
pMultipleResults->GetResult(NULL, 0,
IID_IRowset, &IRowsAffected, (IUnknown
**)&pRowset);
if
(FAILED(hr))
{
ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eGetResult, "OrderStatus()");
}
// Fetch
the result row handle(s)
hr =
pRowset->GetNextRows(DB_NULL_HCHAPTER,
0, cRows, &cRowsObtained, &prghRows);

```

```

        if
(FAILED(hr))
        {
            ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eGetNextRows,
"OrderStatus()");
        }

        m_txn.OrderStatus.o_ol_cnt =
(short)cRowsObtained;

        // Get
the data from multiple rows in this rowset
        for (i =
0; i < m_txn.OrderStatus.o_ol_cnt; ++i)
        {
            // Fetch the actual row data by
handle
            hr = pRowset-
>GetData(rghRows[i],
m_hOrderStatusOutputAccessor,
&m_txn.OrderStatus.OL[i]);

            if (FAILED(hr))
            {
                ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eGetData, "OrderStatus()");
            }
        }

        //
Release row(s)
        hr =
pRowset->ReleaseRows(cRowsObtained,
prghRows, NULL, NULL, NULL);

        //
Release rowset
        hr =
pRowset->Release();

        // Get
the second rowset object
        if
(m_txn.OrderStatus.o_ol_cnt > 0)
        {
            hr = pMultipleResults-
>GetResult(NULL, 0, IID_IRowset,
&RowsAffected, (IUnknown **)&pRowset2);

            if (FAILED(hr))
            {
                ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eGetResult, "OrderStatus()");
            }

            // Fetch the result row handle(s)

```

```

            hr = pRowset2-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows2,
&cRowsObtained2, &prghRows2);
            if (FAILED(hr))
            {
                ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eGetNextRows,
"OrderStatus()");
            }

            // Fetch the actual row data by
handle
            hr = pRowset2-
>GetData(rghRows2,
m_hOrderStatusOutputAccessor2,
&m_txn.OrderStatus);

            if (FAILED(hr))
            {
                ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eGetData, "OrderStatus()");
            }

            // Release row(s)
            hr = pRowset2->Release();
        }

        //
Release the common MultipleResults interface
        pMultipleResults->Release();

        if
(m_txn.OrderStatus.o_ol_cnt == 0)
        {
            throw new CTPCC_OLEDB_ERR(
CTPCC_OLEDB_ERR::ERR_NO_SUCH_ORDER );
        }
        else if
(m_txn.OrderStatus.c_id == 0 &&
m_txn.OrderStatus.c_last[0] == 0)
        {
            throw new CTPCC_OLEDB_ERR(
CTPCC_OLEDB_ERR::ERR_INVALID_CUST );
        }
        else
        {
            m_txn.OrderStatus.exec_status_cod
e = eOK;

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

```

```

        // hit
deadlock; backoff for increasingly longer
Sleep(10 * iTryCount);
    }
}

//
if (iTryCount)
//
throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_R
ETRIED_TRANS, iTryCount);
}

void CTPCC_OLEDB::InitDeliveryParams()
{
    int
i;
    HRESULT
hr;
    wchar_t
wchar_t;
    szName[iMAX_SP_NAME_LEN];
    IAccessor*
pIAccessor;

    const ULONG
nInputParams = 2;
    // input parameters
    const ULONG
nOutputParams = 10;
    // output 1st result set columns

    // Structure to bind in accessor
    DBBINDING
acInputDBBinding[nInputParams];
    DBBINDSTATUS
acInputDBBindStatus[nInputParams
];
    DBBINDING
acOutputDBBinding[nOutputParams
];
    DBBINDSTATUS
acOutputDBBindStatus[nOutputPara
ms];

    // Set command text
    _snwprintf(szName,
sizeof(szName)/sizeof(szName[0]),
L"{call
%stpc_delivery(?,?)", m_szSPPrefix);

    // Create and Prepare a new
command object for Delivery.
    CreateCommand(szName,
&m_pIDeliveryCommand);

    // Describe the consumer buffer by
filling in the array
    // of DBBINDING structures. Each binding
associates
    // a single parameter to the consumer's buffer.
    InitBindings(&acInputDBBinding[0],
nInputParams, eInputParameter);

    i = 0;
    // Delivery parameter 1

```

```

        SetBinding(&acInputDBBinding[i++
], offsetof(DELIVERY_DATA, w_id),
sizeof(m_txn.Delivery.w_id), DBTYPE_I4);

    // Delivery parameter 2
    SetBinding(&acInputDBBinding[i++
], offsetof(DELIVERY_DATA, o_carrier_id),
sizeof(m_txn.Delivery.o_carrier_id), DBTYPE_I2);

    hr = m_pIDeliveryCommand-
>QueryInterface(IID_IAccessor, (void
**)&pIAccessor);
    if (FAILED(hr))
    {
        ThrowError(m_pIDeliveryCommand,
COLEDBERR::eQueryInterface, (void
"InitDeliveryParams()");
    }

    hr = pIAccessor->CreateAccessor(
DBACCESSOR_PARAMETERDATA,
    nInputParams,
    acInputDBBinding,
    sizeof(DELIVERY_DATA),
    &m_hDeliveryInputAccessor,
    acInputDBBindStatus);
    if (FAILED(hr))
    {
        ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitDeliveryParams()");
    }

    m_DeliveryExecuteParams.cParamS
ets = 1;
    m_DeliveryExecuteParams.hAccesso
r = m_hDeliveryInputAccessor;
    m_DeliveryExecuteParams.pData =
&m_txn.Delivery;

    // Now fill the binding information
for result set 1 output columns
    InitBindings(&acOutputDBBinding[0
], nOutputParams, eOutputColumn);

    // Binding for a rowset that may
return more than one row.
    for (i = 0; i < 10; ++i)
    {
        // Delivery output
column 1

        SetBinding(&acOutputDBBinding[i],
offsetof(DELIVERY_DATA, o_id[i]),
sizeof(m_txn.Delivery.o_id[i]), DBTYPE_I4);
    }

    hr = pIAccessor->CreateAccessor(
DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
    nOutputParams,
    acOutputDBBinding,
    sizeof(DELIVERY_DATA),
&m_hDeliveryOutputAccessor,
    acOutputDBBindStatus);
    if (FAILED(hr))
    {
        ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitDeliveryParams()");
    }

```

```

    }
}

void CTPCC_OLEDB::Delivery()
{
    HRESULT
    hr;
    int
    iTryCount = 0;

    IRowset*
    pRowset;
    LONG
    cRows = 1; // number of rows
returned in the rowset
    ULONG
    cRowsObtained;
    HROW
    rghRow;
    //returned row handles
    HROW*
    prghRow = &rghRow;

    while (TRUE)
    {
        try
        {
            //
            Execute the prepared command
            hr =
            m_pIDeliveryCommand->Execute(NULL,
IID_IRowset, &m_DeliveryExecuteParams, NULL,

            (IUnknown
            **)&pRowset);

            if
            (FAILED(hr))
            {
                ThrowError(m_pIDeliveryCommand,
COLEDBERR::eExecute, "Delivery()");
            }

            // Fetch
            the result row handle(s)
            hr =
            pRowset->GetNextRows(DB_NULL_HCHAPTER,
0, cRows, &cRowsObtained, &prghRow);
            if
            (FAILED(hr))
            {
                ThrowError(m_pIDeliveryCommand,
COLEDBERR::eGetNextRows, "Delivery()");
            }

            // Fetch
            the actual row data by handle
            hr =
            pRowset->GetData(rghRow,
m_hDeliveryOutputAccessor, &m_txn.Delivery);
            if
            (FAILED(hr))
            {
                ThrowError(m_pIDeliveryCommand,
COLEDBERR::eGetData, "Delivery()");
            }

            //
            Release row(s)

```

```

            hr =
            pRowset->ReleaseRows(cRowsObtained,
prghRow, NULL, NULL, NULL); //
            Release rowset
            hr =
            pRowset->Release();

            m_txn.Delivery.exec_status_code =
eOK;

            break;
        }
        catch (COLEDBERR
*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_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_R
ETRIED_TRANS, iTryCount);
    }
}

```

**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;
    ULONG ulTmpSize = 0;

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

    m_bSinglePool
    =
    bSinglePool;

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

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

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

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

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

        // all txns will use
        same component
        m_pPayment =
        m_pNewOrder;
        m_pStockLevel =
        m_pNewOrder;
    }
}

```

```

        m_pOrderStatus =
        m_pNewOrder;
        else
        {
            // use different
            components for each txn

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

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

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

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

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

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

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

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

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

```

```

            SafeArrayDestroy(m_vTxn.parray);
            ReleaseInterface(m_pNewOrder);
            if (!m_bSinglePool)
            {
                ReleaseInterface(m_pPayment);

                ReleaseInterface(m_pStockLevel);

                ReleaseInterface(m_pOrderStatus);
            }
            CoUninitialize();
        }

void CTPCC_COM::NewOrder()
{
    VARIANT vTxn_out;

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

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

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

void CTPCC_COM::Payment()
{
    VARIANT vTxn_out;

    HRESULT hr = m_pPayment-
    >Payment(m_vTxn, &vTxn_out);
    if (FAILED(hr))
        throw new CCOMERR(
    hr );

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

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

void CTPCC_COM::StockLevel()
{
    VARIANT vTxn_out;

    HRESULT hr = m_pStockLevel-
    >StockLevel(m_vTxn, &vTxn_out);
    if (FAILED(hr))
        throw new CCOMERR(
    hr );

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

```

```

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

void CTPCC_COM::OrderStatus()
{
    VARIANT vTxn_out;

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

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

```

**\tpcc\_com\_all\src\methods.h**

```

/*      FILE:
      METHODS.H

      *
      *      Microsoft TPC-C Kit Ver. 4.20.000

      *
      *      Copyright Microsoft, 1999
      *      All Rights Reserved
      *
      *
      *      not yet audited
      *
      *      PURPOSE:  Header file for COM
components.
      *
      *      Change history:
      *
      *          4.20.000 - first
version
      */

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

class CCOMPONENT_ERR : public CBaseErr
{
public:

    CCOMPONENT_ERR(COMPONENT_
ERROR Err)
        {
            m_Error

= Err;

            m_szTextDetail = NULL;

```

```

        m_SystemErr = 0;
        m_szErrorText = NULL;
        };

        CCOMPONENT_ERR(COMPONENT_
ERROR Err, char *szTextDetail, DWORD
dwSystemErr)
        {
            m_Error

= Err;

            m_szTextDetail = new
char[strlen(szTextDetail)+1];
            strcpy(
m_szTextDetail, szTextDetail );

            m_SystemErr = dwSystemErr;
            m_szErrorText = NULL;
        };

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

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

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

        static void WriteMessageToEventLog(LPTSTR
lpzMsg);

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

```

```

            COM_INTERFACE_ENTRY(IObjectC
ontrol)
            COM_INTERFACE_ENTRY(IObjectC
onstruct)
            END_COM_MAP()

            CTPCC_Common();
            ~CTPCC_Common();

        // ITPCC
public:
            HRESULT __stdcall NewOrder(
                VARIANT txn_in,
                VARIANT* txn_out);
            HRESULT __stdcall Payment(
                VARIANT txn_in,
                VARIANT* txn_out);
            HRESULT __stdcall Delivery(
                VARIANT txn_in,
                VARIANT* txn_out) {return E_NOTIMPL;};
            HRESULT __stdcall StockLevel(
                VARIANT txn_in, VARIANT*
txn_out);
            HRESULT __stdcall OrderStatus(
                VARIANT txn_in, VARIANT*
txn_out);

            HRESULT __stdcall
CallSetComplete();

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

        // IObjectConstruct
        STDMETHODIMP
Construct(IDispatch * pUnk);

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

        struct COM_DATA
        {
            int retval;
            int error;
            union
            {
                NEW_ORDER_DATA
                NewOrder;

                PAYMENT_DATA
                Payment;

                DELIVERY_DATA
                Delivery;

                STOCK_LEVEL_DATA
                StockLevel;

                ORDER_STATUS_DATA
                OrderStatus;
            } u;
        };
};

```

```

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

BEGIN_COM_MAP(CTPCC)
    //COM_INTERFACE_ENTRY2(IUnkno
own,
CComObjectRootEx<CComSingleThreadModel>)
    COM_INTERFACE_ENTRY2(IUnkno
wn, ITPCC)
    COM_INTERFACE_ENTRY_CHAIN(C
TPCC_Common)
END_COM_MAP()
};

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

BEGIN_COM_MAP(CNewOrder)
//
    COM_INTERFACE_ENTRY2(IUnkno
wn, CComObjectRootEx)
    COM_INTERFACE_ENTRY2(IUnkno
wn, ITPCC)
    COM_INTERFACE_ENTRY_CHAIN(C
TPCC_Common)
END_COM_MAP()

// ITPCC
public:
//
    HRESULT __stdcall NewOrder(
        VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
    HRESULT __stdcall Payment(
        VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
    HRESULT __stdcall StockLevel(
        VARIANT txn_in, VARIANT*
txn_out) {return E_NOTIMPL;}
    HRESULT __stdcall OrderStatus(
        VARIANT txn_in, VARIANT*
txn_out) {return E_NOTIMPL;}
};

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

```

```

BEGIN_COM_MAP(COrderStatus)
//
    COM_INTERFACE_ENTRY2(IUnkno
wn, CComObjectRootEx)
    COM_INTERFACE_ENTRY2(IUnkno
wn, ITPCC)
    COM_INTERFACE_ENTRY_CHAIN(C
TPCC_Common)
END_COM_MAP()

// ITPCC
public:
    HRESULT __stdcall NewOrder(
        VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
    HRESULT __stdcall Payment(
        VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
    HRESULT __stdcall StockLevel(
        VARIANT txn_in, VARIANT*
txn_out) {return E_NOTIMPL;}
//
    HRESULT __stdcall OrderStatus(
        VARIANT txn_in, VARIANT*
txn_out) {return E_NOTIMPL;}
};

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

BEGIN_COM_MAP(CPayment)
//
    COM_INTERFACE_ENTRY2(IUnkno
wn, CComObjectRootEx)
    COM_INTERFACE_ENTRY2(IUnkno
wn, ITPCC)
    COM_INTERFACE_ENTRY_CHAIN(C
TPCC_Common)
END_COM_MAP()

// ITPCC
public:
    HRESULT __stdcall NewOrder(
        VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
//
    HRESULT __stdcall Payment(
        VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
    HRESULT __stdcall StockLevel(
        VARIANT txn_in, VARIANT*
txn_out) {return E_NOTIMPL;}
    HRESULT __stdcall OrderStatus(
        VARIANT txn_in, VARIANT*
txn_out) {return E_NOTIMPL;}
};

////////////////////////////////////
////////////////////////////////////
// CStockLevel
class CStockLevel :
    public CTPCC_Common,
    public CComCoClass<CStockLevel,
&CLSID_StockLevel>
{
public:

```

```

DECLARE_REGISTRY_RESOURCEID(IDR_STOCKL
EVEL)

BEGIN_COM_MAP(CStockLevel)
//
    COM_INTERFACE_ENTRY2(IUnkno
wn, CComObjectRootEx)
    COM_INTERFACE_ENTRY2(IUnkno
wn, ITPCC)
    COM_INTERFACE_ENTRY_CHAIN(C
TPCC_Common)
END_COM_MAP()

// ITPCC
public:
    HRESULT __stdcall NewOrder(
        VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
    HRESULT __stdcall Payment(
        VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
//
    HRESULT __stdcall StockLevel(
        VARIANT txn_in, VARIANT*
txn_out) {return E_NOTIMPL;}
    HRESULT __stdcall OrderStatus(
        VARIANT txn_in, VARIANT*
txn_out) {return E_NOTIMPL;}
};

```

```

\tpcc_com_all\src\resource.
h

```

```

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

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

```

```

\tpcc_com_all\src\tpcc_com
all.cpp

```

```

/*      FILE:
        TPCC_COM_ALL.CPP
*
        Microsoft TPC-C Kit Ver. 4.20.000

```

```

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

#define STRICT
#define _WIN32_WINNT 0x0400
#define _ATL_APARTMENT_THREADED

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

#include <atcom.h>
#include <initguid.h>
#include <transact.h>
//#include <atimpl.cpp>
#include <comsvcs.h>

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

#include "tpcc_com_ps.h"
#include "..\..\common\src\trans.h"

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

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

CComModule _Module;

BEGIN_OBJECT_MAP(ObjectMap)
OBJECT_ENTRY(CLSID_TPCC,
CTPCC)
OBJECT_ENTRY(CLSID_NewOrder,
CNewOrder)
OBJECT_ENTRY(CLSID_OrderStatus
, COrderStatus)
OBJECT_ENTRY(CLSID_Payment,
CPayment)

```

```

OBJECT_ENTRY(CLSID_StockLevel,
CStockLevel)_MAP()

// configuration settings from registry
TPCCREGISTRYDATA Reg;
char
szMyComputerName[MAX_COMPUT
ERNAME_LENGTH+1];

static HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB
* pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC
* pCTPCC_ODBC_new;

// Critical section to synchronize connection open
and close.
//
CRITICAL_SECTION hConnectCriticalSection;

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

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

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

DisableThreadLibraryCalls(hInstanc
e);

DWORD
dwSize = MAX_COMPUTERNAME_LENGTH+1;

GetComputerName(szMyComputerN
ame, &dwSize);

szMyComputerName[dwSize] = 0;

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

if
(Reg.eDB_Protocol == DBLIB)
{
strcpy( szDllName, Reg.szPath );

strcat( szDllName, "tpcc_dblib.dll");

hLibInstanceDb = LoadLibrary(
szDllName );

if (hLibInstanceDb == NULL)
throw new
CCOMPONENT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );
}
}
}
}

```

```

// get function pointer to wrapper
for class constructor

pCTPCC_DBLIB_new =
(TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_
new");

if (pCTPCC_DBLIB_new == NULL)

throw new
CCOMPONENT_ERR(
ERR_GETPROCADDR_FAILED, szDllName,
GetLastError() );
}
else if
(Reg.eDB_Protocol == ODBC)
{
strcpy( szDllName, Reg.szPath );

strcat( szDllName, "tpcc_odbc.dll");

hLibInstanceDb = LoadLibrary(
szDllName );

if (hLibInstanceDb == NULL)

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

// get function pointer to wrapper
for class constructor

pCTPCC_ODBC_new =
(TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_
new");

if (pCTPCC_ODBC_new == NULL)

throw new
CCOMPONENT_ERR(
ERR_GETPROCADDR_FAILED, szDllName,
GetLastError() );
}
else
throw new CCOMPONENT_ERR(
ERR_UNKNOWN_DB_PROTOCOL );

if
(Reg.dwConnectDelay > 0)
{
InitializeCriticalSection(&hConnectC
riticalSection);
}
}
else if (dwReason ==
DLL_PROCESS_DETACH)
_Module.Term();
}
catch (CBaseErr *e)
{
TCHAR szMsg[256];

```

```

        _sntprintf(szMsg,
sizeof(szMsg), "%s error, code %d: %s",
e->ErrorTypeStr(), e->ErrorNum(),
e->ErrorText());

        WriteMessageToEventLog( szMsg );

        delete e;
        return FALSE;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("U
nhandled exception in object DllMain"));
        return FALSE;
    }

    return TRUE; // OK
}

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

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

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

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

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

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

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

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

static void WriteMessageToEventLog(LPTSTR
lpszMsg)
{
    TCHAR szMsg[256];

```

```

        HANDLE hEventSource;

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

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

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

            (VOID)
DeregisterEventSource(hEventSource);
        }

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

        /* FUNCTION: CCOMPONENT_ERR::ErrorText
        *
        */

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

```

```

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

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

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

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

    CTPCC_Common::~~CTPCC_Common()
    {
        // Pace connection close for VIA.
        //
        if (Reg.dwConnectDelay > 0)
        {
            EnterCriticalSection(&hConnectCriti
calSection);

            Sleep(Reg.dwConnectDelay);
        }

        if (m_pTxn)
        {
            delete m_pTxn;
        }

        if (Reg.dwConnectDelay > 0)
        {
            LeaveCriticalSection(&hConnectCrite
calSection);
        }
    }

    HRESULT CTPCC_Common::CallSetComplete()
    {

```



```

IObjectContext* pObjContext =
NULL;

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

//
// called by the ctor activator

//
// STDMETHODCALLTYPE
CTPCC_Common::Construct(IDispatch * pUnk)
{
// Code to access construction
string, if needed later...
// if (!pUnk)
// return
E_UNEXPECTED;
//
IObjectContext * pString =
NULL;
// HRESULT hr = pUnk-
>QueryInterface(IID_IObjectContextString,
(void **) &pString);
// pString->Release();

try
{
// Pace connection
creation for VIA.
//
// if
(Reg.dwConnectDelay > 0)
{
EnterCriticalSection(&hConnectCriti
calSection);

Sleep(Reg.dwConnectDelay);
}

if (Reg.eDB_Protocol
== ODBC)
m_pTxn
= pCTPCC_ODBC_new( Reg.szDbServer,
Reg.szDbUser, Reg.szDbPassword,

szMyComputerName,
Reg.szDbName,

Reg.szSPPrefix,
Reg.bCallNoDuplicatesNewOrder );
else if
(Reg.eDB_Protocol == DBLIB)
m_pTxn
= pCTPCC_DBLIB_new( Reg.szDbServer,
Reg.szDbUser, Reg.szDbPassword,
szMyComputerName, Reg.szDbName );

if
(Reg.dwConnectDelay > 0)
{

```

```

LeaveCriticalSection(&hConnectCriti
calSection);
}
}
catch (CBaseErr *e)
{
TCHAR szMsg[256];

_sntprintf(szMsg,
sizeof(szMsg), "%s error in
CTPCC_Common::Construct, code %d: %s",
e->ErrorTypeStr(), e->ErrorNum(),
e->ErrorText());

WriteMessageToEventLog( szMsg );
delete e;
return E_FAIL;
}
catch (...)
{
WriteMessageToEventLog(TEXT("U
nhandled exception in object ::Construct"));
return E_FAIL;
}

return S_OK;
}

HRESULT CTPCC_Common::NewOrder(VARIANT
txn_in, VARIANT* txn_out)
{
PNEW_ORDER_DATA
pNewOrder;
COM_DATA *pData;
try
{
pData =
(COM_DATA*)txn_in.parray->pvData;
pNewOrder =
m_pTxn->BuffAddr_NewOrder();

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

m_pTxn-
>NewOrder(); // do the
actual txn

VariantInit(txn_out);
txn_out->vt =
VT_SAFEARRAY;
txn_out->parray =
SafeArrayCreateVector(VT_UI1,

txn_in.parray->rgsabound-
>cElements,

txn_in.parray->rgsabound-
>cElements);

pData =
(COM_DATA*) txn_out->parray->pvData;

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

pData->retval =
ERR_SUCCESS;

pData->error = 0;
return S_OK;
}

```

```

catch (CBaseErr *e)
{
// check for lost
database connection; if yes, component is toast
if ( ((e->ErrorType()
== ERR_TYPE_DBLIB) && (e->ErrorNum() ==
10005)) ||
((e-
>ErrorType() == ERR_TYPE_ODBC) && (e-
>ErrorNum() == 10054)) )

m_bCanBePooled = FALSE;

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

HRESULT CTPCC_Common::Payment(VARIANT
txn_in, VARIANT* txn_out)
{
PPAYMENT_DATA
pPayment;
COM_DATA *pData;
try
{
pData =
(COM_DATA*)txn_in.parray->pvData;
pPayment = m_pTxn-
>BuffAddr_Payment();

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

m_pTxn->Payment();
// do the actual txn

VariantInit(txn_out);
txn_out->vt =
VT_SAFEARRAY;
txn_out->parray =
SafeArrayCreateVector( VT_UI1,

txn_in.parray->rgsabound-
>cElements,

txn_in.parray->rgsabound-
>cElements);

pData =
(COM_DATA*) txn_out->parray->pvData;

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

```

```

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

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

HRESULT CTPCC_Common::StockLevel(VARIANT
txn_in, VARIANT* txn_out)
{
    PSTOCK_LEVEL_DATA
pStockLevel;
    COM_DATA
*pData;

    try
    {
        pData =
(COM_DATA*)txn_in.parray->pvData;
        pStockLevel =
m_pTxn->BuffAddr_StockLevel();

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

        m_pTxn-
>StockLevel();

        VariantInit(txn_out);
        txn_out->vt =
VT_SAFEARRAY;
        txn_out->parray =
SafeArrayCreateVector( VT_UI1,

        txn_in.parray->rgsbound-
>cElements,

        txn_in.parray->rgsbound-
>cElements);

```

```

        pData =
(COM_DATA*)txn_out->parray->pvData;
        memcpy( &pData-
>u.StockLevel, pStockLevel,
sizeof(STOCK_LEVEL_DATA));

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

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

HRESULT
CTPCC_Common::OrderStatus(VARIANT txn_in,
VARIANT* txn_out)
{
    PORDER_STATUS_DATA
pOrderStatus;
    COM_DATA
*pData;
    try
    {
        pData =
(COM_DATA*)txn_in.parray->pvData;
        pOrderStatus =
m_pTxn->BuffAddr_OrderStatus();

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

        m_pTxn-
>OrderStatus();

        VariantInit(txn_out);
        txn_out->vt =
VT_SAFEARRAY;
        txn_out->parray =
SafeArrayCreateVector( VT_UI1,

```

```

        txn_in.parray->rgsbound-
>cElements,

        txn_in.parray->rgsbound-
>cElements);
        pData =
(COM_DATA*)txn_out->parray->pvData;
        memcpy( &pData-
>u.OrderStatus, pOrderStatus,
sizeof(ORDER_STATUS_DATA));

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

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

```

```

\tpcc_com_all\src\tpcc_com_
all.def

```

```

/* FILE:
TPCC_COM_ALL.CPP
*
Microsoft TPC-C Kit Ver. 4.20.000
*
Copyright Microsoft, 1999
All Rights Reserved
*
Version 4.10.000 audited by
Richard Gimarc, Performance Metrics, 3/17/99
*

```

```

*      PURPOSE:  Implementation for
TPC-C Tuxedo class.
*      Contact:  Charles Levine
(clevine@microsoft.com)
*
* Change history:
*      4.20.000 - updated
rev number to match kit
*/

#define STRICT
#define _WIN32_WINNT 0x0400
#define _ATL_APARTMENT_THREADED

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

#include <atcom.h>
#include <initguid.h>
#include <transact.h>
//#include <atimpl.cpp>
#include <comsvcs.h>

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

#include "tpcc_com_ps.h"
#include "..\..\common\src\trans.h"

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

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

CComModule _Module;

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

// configuration settings from registry
TPCCREGISTRYDATA Reg;

```

```

char
szMyComputerName[MAX_COMPUT
ERNAME_LENGTH+1];
static HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB
*pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC
*pCTPCC_ODBC_new;

// Critical section to synchronize connection open
and close.
//
// CRITICAL_SECTION hConnectCriticalSection;

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

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

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

DisableThreadLibraryCalls(hInstanc
e);

DWORD
dwSize = MAX_COMPUTERNAME_LENGTH+1;

GetComputerName(szMyComputerN
ame, &dwSize);

szMyComputerName[dwSize] = 0;

if (
ReadTPCCRegistrySettings( &Reg ) )

throw new CCOMPONENT_ERR(
ERR_MISSING_REGISTRY_ENTRIES);

if
(Reg.eDB_Protocol == DBLIB)
{
strcpy( szDllName, Reg.szPath );

strcat( szDllName, "tpcc_dblib.dll");

hLibInstanceDb = LoadLibrary(
szDllName );

if (hLibInstanceDb == NULL)

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

// get function pointer to wrapper
for class constructor

```

```

pCTPCC_DBLIB_new =
(TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_
new");

if (pCTPCC_DBLIB_new == NULL)

throw new
CCOMPONENT_ERR(
ERR_GETPROCADDR_FAILED, szDllName,
GetLastError() );

}
else if
(Reg.eDB_Protocol == ODBC)
{

strcpy( szDllName, Reg.szPath );

strcat( szDllName, "tpcc_odbc.dll");

hLibInstanceDb = LoadLibrary(
szDllName );

if (hLibInstanceDb == NULL)

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

// get function pointer to wrapper
for class constructor

pCTPCC_ODBC_new =
(TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_
new");

if (pCTPCC_ODBC_new == NULL)

throw new
CCOMPONENT_ERR(
ERR_GETPROCADDR_FAILED, szDllName,
GetLastError() );

}
else
throw new CCOMPONENT_ERR(
ERR_UNKNOWN_DB_PROTOCOL);

if
(Reg.dwConnectDelay > 0)
{

InitializeCriticalSection(&hConnectC
riticalSection);

}

}
else if (dwReason ==
DLL_PROCESS_DETACH)

_Module.Term();

}
catch (CBaseErr *e)
{
TCHAR szMsg[256];

_sntprintf(szMsg,
sizeof(szMsg), "%s error, code %d: %s",

```

```

        e->ErrorTypeStr(), e->ErrorNum(),
        e->ErrorText());

        WriteMessageToEventLog( szMsg );

        delete e;
        return FALSE;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("U
        nhandled exception in object DllMain"));
        return FALSE;
    }

    return TRUE; // OK
}

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

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

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

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

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

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

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

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

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

```

```

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

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

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

        (VOID)
        DeregisterEventSource(hEventSource);
    }

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

    /* FUNCTION: CCOMPONENT_ERR::ErrorText
    *
    */

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

```

```

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

        if (m_szTextDetail)
            strcat( szTmp,
            m_szTextDetail );

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

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

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

    CTPCC_Common::~~CTPCC_Common()
    {
        // Pace connection close for VIA.
        //
        if (Reg.dwConnectDelay > 0)
        {
            EnterCriticalSection(&hConnectCriti
            calSection);

            Sleep(Reg.dwConnectDelay);
        }

        if (m_pTxn)
        {
            delete m_pTxn;
        }

        if (Reg.dwConnectDelay > 0)
        {
            LeaveCriticalSection(&hConnectCritic
            alSection);
        }
    }

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

```

```

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

    //
    // called by the ctor activator

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

        try
        {
            // Pace connection
            creation for VIA.

            //
            if
            (Reg.dwConnectDelay > 0)
            {
                EnterCriticalSection(&hConnectCriti
                calSection);

                Sleep(Reg.dwConnectDelay);
            }

            if (Reg.eDB_Protocol
            == ODBC)
            {
                m_pTxn
                = pCTPCC_ODBC_new( Reg.szDbServer,
                Reg.szDbUser, Reg.szDbPassword,

                szMyComputerName,
                Reg.szDbName,

                Reg.szSPPrefix,
                Reg.bCallNoDuplicatesNewOrder );
            }
            else if
            (Reg.eDB_Protocol == DBLIB)
            {
                m_pTxn
                = pCTPCC_DBLIB_new( Reg.szDbServer,
                Reg.szDbUser, Reg.szDbPassword,
                szMyComputerName, Reg.szDbName );
            }

            if
            (Reg.dwConnectDelay > 0)
            {
                LeaveCriticalSection(&hConnectCrite
                calSection);
            }
        }
    }

```

```

        catch (CBaseErr *e)
        {
            TCHAR szMsg[256];
            _sntprintf(szMsg,
            sizeof(szMsg), "%s error in
            CTPCC_Common::Construct, code %d: %s",
            e->ErrorTypeStr(), e->ErrorNum(),
            e->ErrorText());

            WriteMessageToEventLog( szMsg );
            delete e;
            return E_FAIL;
        }
        catch (...)
        {
            WriteMessageToEventLog(TEXT("U
            nhandled exception in object ::Construct"));
            return E_FAIL;
        }
    }

    return S_OK;
}

HRESULT CTPCC_Common::NewOrder(VARIANT
txn_in, VARIANT* txn_out)
{
    PNEW_ORDER_DATA
    pNewOrder;
    COM_DATA *pData;
    try
    {
        pData =
        (COM_DATA*)txn_in.parray->pvData;
        pNewOrder =
        m_pTxn->BuffAddr_NewOrder();

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

        m_pTxn-
        >NewOrder(); // do the
        actual txn

        VariantInit(txn_out);
        txn_out->vt =
        VT_SAFEARRAY;
        txn_out->parray =
        SafeArrayCreateVector(VT_UI1,

        txn_in.parray->rgsabound-
        >cElements,

        txn_in.parray->rgsabound-
        >cElements);

        pData =
        (COM_DATA*) txn_out->parray->pvData;

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

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

```

```

        if ( ((e->ErrorType()
        == ERR_TYPE_DBLIB) && (e->ErrorNum() ==
        10005)) || ((e-
        >ErrorType() == ERR_TYPE_ODBC) && (e-
        >ErrorNum() == 10054)) )

            m_bCanBePooled = FALSE;

            pData->retval = e-
            >ErrorType();
            pData->error = e-
            >ErrorNum();

            delete e;
            return E_FAIL;
        }
        catch (...)
        {
            WriteMessageToEventLog(TEXT("U
            nhandled exception.));

            pData->retval =
            ERR_TYPE_LOGIC;

            pData->error = 0;
            m_bCanBePooled =
            FALSE;

            return E_FAIL;
        }
    }

    HRESULT CTPCC_Common::Payment(VARIANT
    txn_in, VARIANT* txn_out)
    {
        PPAYMENT_DATA
        pPayment;
        COM_DATA *pData;
        try
        {
            pData =
            (COM_DATA*)txn_in.parray->pvData;
            pPayment = m_pTxn-
            >BuffAddr_Payment();

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

            m_pTxn->Payment();
            // do the actual txn

            VariantInit(txn_out);
            txn_out->vt =
            VT_SAFEARRAY;
            txn_out->parray =
            SafeArrayCreateVector( VT_UI1,

            txn_in.parray->rgsabound-
            >cElements,

            txn_in.parray->rgsabound-
            >cElements);

            pData =
            (COM_DATA*) txn_out->parray->pvData;

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

            pData->retval =
            ERR_SUCCESS;
            pData->error = 0;
            return S_OK;
        }
    }
}

```

```

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

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

    HRESULT CTPCC_Common::StockLevel(VARIANT
    txn_in, VARIANT* txn_out)
    {
        PSTOCK_LEVEL_DATA
        pStockLevel;
        COM_DATA
        *pData;

        try
        {
            pData =
            (COM_DATA*)txn_in.parray->pvData;
            pStockLevel =
            m_pTxn->BuffAddr_StockLevel();

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

            m_pTxn-
            >StockLevel();

            VariantInit(txn_out);
            txn_out->vt =
            VT_SAFEARRAY;
            txn_out->parray =
            SafeArrayCreateVector( VT_UI1,

            txn_in.parray->rgsabound-
            >cElements,

            txn_in.parray->rgsabound-
            >cElements);
            pData =
            (COM_DATA*)txn_out->parray->pvData;

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

```

```

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

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

    HRESULT
    CTPCC_Common::OrderStatus(VARIANT txn_in,
    VARIANT* txn_out)
    {
        PORDER_STATUS_DATA
        pOrderStatus;
        COM_DATA
        *pData;
        try
        {
            pData =
            (COM_DATA*)txn_in.parray->pvData;
            pOrderStatus =
            m_pTxn->BuffAddr_OrderStatus();

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

            m_pTxn-
            >OrderStatus();

            VariantInit(txn_out);
            txn_out->vt =
            VT_SAFEARRAY;
            txn_out->parray =
            SafeArrayCreateVector( VT_UI1,

            txn_in.parray->rgsabound-
            >cElements,

            txn_in.parray->rgsabound-
            >cElements);

            txn_in.parray->rgsabound-
            >cElements);

```

```

            pData =
            (COM_DATA*)txn_out->parray->pvData;
            memcpy( &pData-
            >u.OrderStatus, pOrderStatus,
            sizeof(ORDER_STATUS_DATA));

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

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

```

#### \tpcc\_com\_all.h

```

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

```

```

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

```

```

/* File created by MIDL compiler version
6.00.0347 */
/* at Fri Apr 15 14:48:53 2005
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
protocol : dce , ms_ext, c_ext
error checks: allocation ref bounds_check
enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany),
__declspec(notable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/

```

```

//@@MIDL_FILE_HEADING( )

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

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

#ifndef __tpcc_com_all_h__
#define __tpcc_com_all_h__

#if defined(_MSC_VER) && (_MSC_VER >=
1020)
#pragma once
#endif

/* Forward Declarations */

#ifndef __TPCC_FWD_DEFINED__
#define __TPCC_FWD_DEFINED__

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

#endif /* __TPCC_FWD_DEFINED__ */

#ifndef __NewOrder_FWD_DEFINED__
#define __NewOrder_FWD_DEFINED__

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

#endif /* __NewOrder_FWD_DEFINED__ */

#ifndef __OrderStatus_FWD_DEFINED__
#define __OrderStatus_FWD_DEFINED__

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

#endif /* __OrderStatus_FWD_DEFINED__ */

#ifndef __Payment_FWD_DEFINED__
#define __Payment_FWD_DEFINED__

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

#endif /* __Payment_FWD_DEFINED__ */

#ifndef __StockLevel_FWD_DEFINED__
#define __StockLevel_FWD_DEFINED__

```

```

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

#endif /* __StockLevel_FWD_DEFINED__ */

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

#ifdef __cplusplus
extern "C"{
#endif

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

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

extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_all_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_all_0000_v0_0_s_ifspec;

#ifdef __TPCCLib_LIBRARY_DEFINED__
#define __TPCCLib_LIBRARY_DEFINED__

/* library TPCCLib */
/* [helpstring][version][uuid] */

EXTERN_C const IID LIBID_TPCCLib;

EXTERN_C const CLSID CLSID_TPCC;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_NewOrder;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_OrderStatus;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_Payment;

```

```

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_StockLevel;

#ifdef __cplusplus

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

/* Additional Prototypes for ALL interfaces */

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

```

```

\tpcc_com_all\src\tpcc_com
all.idl

```

```

/* FILE:
TPCC.IDL

Microsoft TPC-C Kit Ver. 4.20.000

Copyright Microsoft, 1999
All Rights Reserved

not yet audited

PURPOSE: IDL source for
TPCC.dll. This file is processed by the MIDL tool
to
produce the type library (TPCC.tlb)
and marshalling code.

* Change history:
* 4.20.000 - first
version
*/

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

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

[
    uuid(122A3117-2520-11D3-BA71-
00C04FBFE08B),
    version(1.0),

```

```

Library")
]
library TPCCLib
{
    importlib("stdole32.tlb");
    importlib("stdole2.tlb");

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

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

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

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

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

```

```

\tpcc_com_all\src\tpcc_com_
all.rc

```

```

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

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

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

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

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

#ifdef APSTUDIO_INVOKED
////////////////////////////////////
////////////////////////////////////
//
// TEXTINCLUDE
//
1 TEXTINCLUDE DISCARDABLE
BEGIN
    "resource.h\0"
END

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

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

#endif // APSTUDIO_INVOKED

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

VS_VERSION_INFO VERSIONINFO
FILEVERSION 1,0,0,1
PRODUCTVERSION 1,0,0,1
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L

```

```

#ifdef _WIN32
FILEOS 0x4L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904B0"
        BEGIN
            VALUE "CompanyName", "\0"
            VALUE "FileDescription", "tpcc_com_all
Module\0"
            VALUE "FileVersion", "1, 0, 0, 1\0"
            VALUE "InternalName",
"TPCCNEWORDER\0"
            VALUE "LegalCopyright", "Copyright
1997\0"
            VALUE "OriginalFilename",
"tpcc_com_all.DLL\0"
            VALUE "ProductName", "tpcc_com_all
Module\0"
            VALUE "ProductVersion", "1, 0, 0, 1\0"
            VALUE "OLESelfRegister", "\0"
        END
    END
    BLOCK "VarFileInfo"

    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END

#endif // !_MAC

////////////////////////////////////
////////////////////////////////////
//
// REGISTRY
//

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

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

STRINGTABLE DISCARDABLE
BEGIN
    IDS_PROJNAME "tpcc_com_all"
END

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

////////////////////////////////////
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//

#ifdef APSTUDIO_INVOKED
////////////////////////////////////
////////////////////////////////////
//
//
// Generated from the TEXTINCLUDE 3 resource.
//

```



```
1 TYPELIB "tpcc_com_all.tib"

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
//endf // not APSTUDIO_INVOKED
```

```
\tpcc_com_all\src\tpcc_com_all.rgs
```

```
HKCR
{
    TPCC.AllTxns.1 = s 'All Txns Class'
    {
        CLSID = s
        '{122A3128-2520-11D3-BA71-00C04FBE08B}'
    }
    TPCC.AllTxns = s 'TPCC Class'
    {
        CurVer = s
        'TPCC.AllTxns.1'
    }
    NoRemove CLSID
    {
        ForceRemove
        '{122A3128-2520-11D3-BA71-00C04FBE08B}' =
        s 'TPCC Class'
    }
    ProgID
    = s 'TPCC.AllTxns.1'

    VersionIndependentProgID = s
    'TPCC.AllTxns'

    InprocServer32 = s '%MODULE%'
    {
        val ThreadingModel = s 'Both'
    }
}
}
```

```
\tpcc_com_all\src\tpcc_com_all_i.c
```

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

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

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

/* File created by MIDL compiler version
6.00.0347 */
/* at Fri Apr 15 14:48:53 2005
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
protocol : dce , ms_ext, c_ext
error checks: allocation ref bounds_check
enum stub_data
```

```
VC__declspec(selectany),
__declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADERING( )

#if !defined(_M_IA64) && !defined(_M_AMD64)

#ifdef __cplusplus
extern "C"{
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

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

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b
3,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;

#ifdef __IID_DEFINED__

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

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

#endif // !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCClib,0x122A3117,0x2520,0x11D3,0xB
A,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

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

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

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

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

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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#ifdef /* !defined(_M_IA64) &&
!defined(_M_AMD64)*/

#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
6.00.0347 */
/* at Fri Apr 15 14:48:53 2005
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf, W1, Zp8, env=Win64 (32b
run,appending)
protocol : dce , 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_HEADERING( )

#if defined(_M_IA64) || defined(_M_AMD64)

#ifdef __cplusplus
extern "C"{
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

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

```

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,
3,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,
3,b4,b5,b6,b7,b8) \
    const type name =
{l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

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

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

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

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

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

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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif /* defined(_M_IA64) ||
defined(_M_AMD64)*/

```

```

\tpcc_com_all\src\tpcc_com
no.rgs

```

```

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

```

```

\tpcc_com_all\src\tpcc_com
os.rgs

```

```

HKCR
{
    TPCC.OrderStatus.1 = s
'OrderStatus Class'
    {
        CLSID = s
        '{266836AD-A50D-11D2-BA4E-00C04FBFE08B}'
    }
    TPCC.OrderStatus = s 'OrderStatus
Class'
    {
        CurVer = s
        'TPCC.OrderStatus.1'
    }
    NoRemove CLSID
    {
        ForceRemove
        {266836AD-A50D-11D2-BA4E-00C04FBFE08B} =
s 'OrderStatus Class'
    }
    ProgID
    = s 'TPCC.OrderStatus.1'
    VersionIndependentProgID = s
    'TPCC.OrderStatus'

```

```

InprocServer32 = s '%MODULE%'
    val ThreadingModel = s 'Both'
    }
}

```

```

\tpcc_com_all\src\tpcc_com
pay.rgs

```

```

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

```

```

\tpcc_com_all\src\tpcc_com
ps.h

```

```

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

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

/* File created by MIDL compiler version
6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
*/
/* Compiler settings for \src\tpcc_com_ps.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
protocol : dce , 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_HEADER( )

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

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

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

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

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

#if defined(_MSC_VER) && (_MSC_VER >=
1020)
#pragma once
#endif

/* Forward Declarations */

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

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

#ifdef __cplusplus
extern "C"{
#endif

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

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

extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

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

```

```

EXTERN_C const IID IID_ITPCC;

#ifdef __cplusplus &&
!defined(CINTERFACE)

MIDL_INTERFACE("FEE6AA2-84B1-11d2-
BA47-00C04FBFE08B")
ITPCC : public IUnknown
{
public:
virtual HRESULT STDMETHODCALLTYPE NewOrder(
/* [in] */ VARIANT txn_in,
/* [out] */ VARIANT *txn_out) = 0;

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

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

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

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

virtual HRESULT STDMETHODCALLTYPE CallSetComplete(
void) = 0;
};

#else /* C style interface */

typedef struct ITPCCVtbl
{
BEGIN_INTERFACE

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

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

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

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

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

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

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

```

```

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

HRESULT ( STDMETHODCALLTYPE *CallSetComplete )(
ITPCC * This);

END_INTERFACE
} ITPCCVtbl;

interface ITPCC
{
CONST_VTBL struct ITPCCVtbl *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 */

#ifdef /* C style interface */

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

```

```

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

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

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

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

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

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

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

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

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

HRESULT __stdcall
ITPCC_CallSetComplete_Proxy(
    ITPCC * 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 *, unsigned
long                , VARIANT * );
unsigned char * __RPC_USER
VARIANT_UserMarshal( unsigned long *,
unsigned char *, VARIANT * );
unsigned char * __RPC_USER
VARIANT_UserUnmarshal(unsigned long *,
unsigned char *, VARIANT * );
void __RPC_USER
VARIANT_UserFree(    unsigned long *,
VARIANT * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

\tpcc_com_all\src\tpcc_com_sl.rgs

HKCR
{
    TPCC.StockLevel.1 = s 'StockLevel
Class'
    {
        CLSID = s
        '{2668369E-A50D-11D2-BA4E-00C04FBFE08B}'
    }
    TPCC.StockLevel = s 'StockLevel
Class'
    {
        CurVer = s
        'TPCC.StockLevel.1'
    }
    NoRemove CLSID
    {
        ForceRemove
        {2668369E-A50D-11D2-BA4E-00C04FBFE08B} =
        s 'StockLevel Class'
    }
    ProgID
    = s 'TPCC.StockLevel.1'

    VersionIndependentProgID = s
    'TPCC.StockLevel'

    InprocServer32 = s '%MODULE%'
    {
        val ThreadingModel = s 'Both'
    }
}

\tpcc_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 \*/

tpcc\_com\_ps\src\tpcc\_com\_ps  
.def

LIBRARY "tpcc\_com\_ps"

DESCRIPTION 'Proxy/Stub DLL'

EXPORTS  
DllGetObject @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
6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
protocol : dce , ms_ext, c_ext
error checks: allocation ref bounds_check
enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany),
__declspec(notvably)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADERING( )
```

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

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

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

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

```
#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__
```

```
#if defined(_MSC_VER) && (_MSC_VER >=
1020)
#pragma once
#endif
```

```
/* Forward Declarations */
```

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

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

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

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

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

```
extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;
```

```
#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__
```

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

```
EXTERN_C const IID IID_ITPCC;
```

```
#if defined(_cplusplus) &&
!defined(CINTERFACE)
```

```
MIDL_INTERFACE("FEEE6AA2-84B1-11d2-
BA47-00C04FBFE08B")
ITPCC : public IUnknown
{
```

```
public:
virtual HRESULT STDMETHODCALLTYPE NewOrder(
/* [in] */ VARIANT bx_in,
/* [out] */ VARIANT *bx_out) = 0;
```

```
virtual HRESULT STDMETHODCALLTYPE Payment(
/* [in] */ VARIANT bx_in,
/* [out] */ VARIANT *bx_out) = 0;
```

```
virtual HRESULT STDMETHODCALLTYPE Delivery(
/* [in] */ VARIANT bx_in,
/* [out] */ VARIANT *bx_out) = 0;
```

```
virtual HRESULT STDMETHODCALLTYPE StockLevel(
/* [in] */ VARIANT bx_in,
/* [out] */ VARIANT *bx_out) = 0;
```

```
virtual HRESULT STDMETHODCALLTYPE OrderStatus(
/* [in] */ VARIANT bx_in,
/* [out] */ VARIANT *bx_out) = 0;
```

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

```
};
```

```
#else /* C style interface */
```

```
typedef struct ITPCCVtbl
{
```

```
BEGIN_INTERFACE
```

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

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

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

```
HRESULT ( STDMETHODCALLTYPE *NewOrder )(
ITPCC * This,
/* [in] */ VARIANT bx_in,
/* [out] */ VARIANT *bx_out);
```

```
HRESULT ( STDMETHODCALLTYPE *Payment )(
/* [in] */ VARIANT bx_in,
/* [out] */ VARIANT *bx_out);
```

```
HRESULT ( STDMETHODCALLTYPE *Delivery )(
ITPCC * This,
/* [in] */ VARIANT bx_in,
/* [out] */ VARIANT *bx_out);
```

```
HRESULT ( STDMETHODCALLTYPE *StockLevel )(
ITPCC * This,
/* [in] */ VARIANT bx_in,
/* [out] */ VARIANT *bx_out);
```

```
HRESULT ( STDMETHODCALLTYPE *OrderStatus )(
ITPCC * This,
/* [in] */ VARIANT bx_in,
/* [out] */ VARIANT *bx_out);
```

```
HRESULT ( STDMETHODCALLTYPE *CallSetComplete )(
ITPCC * This);
```

```
END_INTERFACE
} ITPCCVtbl;
```

```
interface ITPCC
{
CONST_VTBL struct ITPCCVtbl *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,bx_in,bx_out)
\
(This)->lpVtbl->
NewOrder(This,bx_in,bx_out)
```

```
#define ITPCC_Payment(This,bx_in,bx_out)
\
(This)->lpVtbl->
Payment(This,bx_in,bx_out)
```

```
#define ITPCC_Delivery(This,bx_in,bx_out)
\
(This)->lpVtbl-> Delivery(This,bx_in,bx_out)
```

```
#define ITPCC_StockLevel(This,bx_in,bx_out)
\
(This)->lpVtbl->
StockLevel(This,bx_in,bx_out)
```

```
#define ITPCC_OrderStatus(This,bx_in,bx_out)
\
(This)->lpVtbl->
OrderStatus(This,bx_in,bx_out)
```

```
#define ITPCC_CallSetComplete(This) \
```

```

(This)->lpVtbl -> CallSetComplete(This)
#endif /* COBJMACROS */

#endif /* C style interface */

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

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

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

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

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

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

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

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

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

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

```

```

HRESULT __stdcall
ITPCC_CallSetComplete_Proxy(

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

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif
#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";

```

```

oleautomation,
uuid(FEE6AA2-84B1-
11d2-BA47-00C04FBF0E8B),
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
6.00.0347 */

```

```

/* at Fri Apr 15 14:48:43 2005
*/

```

```

/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
protocol : dce , 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_AMD64)

```

```

#endif

```

```

#endif

```

```

#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,b
3,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,b
3,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,0x
47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

```

```

#undef MIDL_DEFINE_GUID

```

```

#ifdef __cplusplus
}
#endif

```

```

#endif /* !defined(_M_IA64) &&
!defined(_M_AMD64)*/

```

```

#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
6.00.0347 */

```

```

/* at Fri Apr 15 14:48:43 2005
*/

```

```

/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf, W1, Zp8, env=Win64 (32b
run,appending)
protocol : dce , 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( )

```

```

#endif

```

```

#endif

```

```

#endif

```

```

#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,b
3,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,b
3,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,0x
47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

```

```

#undef MIDL_DEFINE_GUID

```

```

#ifdef __cplusplus
}
#endif

```

```

#endif /* defined(_M_IA64) ||
defined(_M_AMD64)*/

```

```

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
6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
protocol : dce , ms_ext, c_ext
error checks: allocation ref bounds_check
enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany),
__declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADERING( )

#ifndef _M_IA64 && !defined(_M_AMD64)
#define USE_STUBLESS_PROXY

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

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

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 1023
#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;

static RPC_SYNTAX_IDENTIFIER
_RpcTransferSyntax =
{{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,
0x00,0x2B,0x10,0x48,0x60}},{2,0}};

extern const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING
__MIDL_ProcFormatString;

extern const MIDL_STUB_DESC
Object_StubDesc;

```

```

extern const MIDL_SERVER_INFO
RPC_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO
ITPCC_ProxyInfo;

extern const
USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ];

#ifndef _RPC_WIN32__
#error Invalid build platform for this stub.
#endif

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

static const MIDL_PROC_FORMAT_STRING
__MIDL_ProcFormatString =
{
0,
{
/* Procedure NewOrder */

0x33,
/* FC_AUTO_HANDLE */

0x6C,
/* Old Flags: object,
Oi2 */
/* 2 */ NdrFcLong( 0x0 ), /* 0 */
/* 6 */ NdrFcShort( 0x3 ), /* 3 */
/* 8 */ NdrFcShort( 0x1c ), /* x86
Stack size/offset = 28 */
/* 10 */ NdrFcShort( 0x0 ), /* 0 */
/* 12 */ NdrFcShort( 0x8 ), /* 8 */
/* 14 */ 0x7, /* Oi2
Flags: srv must size, clt must size, has return, */
/* 3 */
/* Parameter txn_in */

/* 16 */ NdrFcShort( 0x8b ), /* Flags:
must size, must free, in, by val, */
/* 18 */ NdrFcShort( 0x4 ), /* x86
Stack size/offset = 4 */
/* 20 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */
/* Parameter txn_out */

/* 22 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
/* 24 */ NdrFcShort( 0x14 ), /* x86
Stack size/offset = 20 */
/* 26 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */
/* Return value */

/* 28 */ NdrFcShort( 0x70 ), /* Flags:
out, return, base type, */

```

```

/* 30 */ NdrFcShort( 0x18 ), /* x86
Stack size/offset = 24 */
/* 32 */ 0x0, /*
FC_LONG */

/* 34 */ 0x33, /*
FC_AUTO_HANDLE */

/* Procedure Payment */

/* 36 */ NdrFcLong( 0x0 ), /* 0 */
/* 40 */ NdrFcShort( 0x4 ), /* 4 */
/* 42 */ NdrFcShort( 0x1c ), /* x86
Stack size/offset = 28 */
/* 44 */ NdrFcShort( 0x0 ), /* 0 */
/* 46 */ NdrFcShort( 0x8 ), /* 8 */
/* 48 */ 0x7, /* Oi2
Flags: srv must size, clt must size, has return, */
/* 3 */
/* Parameter txn_in */

/* 50 */ NdrFcShort( 0x8b ), /* Flags:
must size, must free, in, by val, */
/* 52 */ NdrFcShort( 0x4 ), /* x86
Stack size/offset = 4 */
/* 54 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */
/* Parameter txn_out */

/* 56 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
/* 58 */ NdrFcShort( 0x14 ), /* x86
Stack size/offset = 20 */
/* 60 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */
/* Return value */

/* 62 */ NdrFcShort( 0x70 ), /* Flags:
out, return, base type, */
/* 64 */ NdrFcShort( 0x18 ), /* x86
Stack size/offset = 24 */
/* 66 */ 0x8, /*
FC_LONG */

/* 68 */ 0x33, /*
FC_AUTO_HANDLE */

/* Old Flags: object,
Oi2 */
/* 70 */ NdrFcLong( 0x0 ), /* 0 */
/* 74 */ NdrFcShort( 0x5 ), /* 5 */
/* 76 */ NdrFcShort( 0x1c ), /* x86
Stack size/offset = 28 */
/* 78 */ NdrFcShort( 0x0 ), /* 0 */
/* 80 */ NdrFcShort( 0x8 ), /* 8 */
/* 82 */ 0x7, /* Oi2
Flags: srv must size, clt must size, has return, */
/* 3 */
/* Parameter txn_in */

```



```

/* 84 */ NdrFcShort( 0x8b ), /* Flags:
must size, must free, in, by val, */
/* 86 */ NdrFcShort( 0x4 ), /* x86
Stack size/offset = 4 */
/* 88 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */

/* Parameter txn_out */

/* 90 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
/* 92 */ NdrFcShort( 0x14 ), /* x86
Stack size/offset = 20 */
/* 94 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */

/* Return value */

/* 96 */ NdrFcShort( 0x70 ), /* Flags:
out, return, base type, */
/* 98 */ NdrFcShort( 0x18 ), /* x86
Stack size/offset = 24 */
/* 100 */ 0x8, /*
FC_LONG */
/* 0 */
/* Procedure StockLevel */

/* 102 */ 0x33, /*
FC_AUTO_HANDLE */
/* Old Flags: object,
Oi2 */
/* 104 */ NdrFcLong( 0x0 ), /* 0 */
/* 108 */ NdrFcShort( 0x6 ), /* 6 */
/* 110 */ NdrFcShort( 0x1c ), /* x86
Stack size/offset = 28 */
/* 112 */ NdrFcShort( 0x0 ), /* 0 */
/* 114 */ NdrFcShort( 0x8 ), /* 8 */
/* 116 */ 0x7, /* Oi2
Flags: srv must size, dt must size, has return, */
/* 3 */
/* Parameter txn_in */

/* 118 */ NdrFcShort( 0x8b ), /* Flags:
must size, must free, in, by val, */
/* 120 */ NdrFcShort( 0x4 ), /* x86
Stack size/offset = 4 */
/* 122 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */

/* Parameter txn_out */

/* 124 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
/* 126 */ NdrFcShort( 0x14 ), /* x86
Stack size/offset = 20 */
/* 128 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */

/* Return value */

/* 130 */ NdrFcShort( 0x70 ), /* Flags:
out, return, base type, */
/* 132 */ NdrFcShort( 0x18 ), /* x86
Stack size/offset = 24 */
/* 134 */ 0x8, /*
FC_LONG */

```

```

/* 0 */
/* Procedure OrderStatus */

/* 136 */ 0x33, /*
FC_AUTO_HANDLE */
/* Old Flags: object,
Oi2 */
/* 138 */ NdrFcLong( 0x0 ), /* 0 */
/* 142 */ NdrFcShort( 0x7 ), /* 7 */
/* 144 */ NdrFcShort( 0x1c ), /* x86
Stack size/offset = 28 */
/* 146 */ NdrFcShort( 0x0 ), /* 0 */
/* 148 */ NdrFcShort( 0x8 ), /* 8 */
/* 150 */ 0x7, /* Oi2
Flags: srv must size, dt must size, has return, */
/* 3 */
/* Parameter txn_in */

/* 152 */ NdrFcShort( 0x8b ), /* Flags:
must size, must free, in, by val, */
/* 154 */ NdrFcShort( 0x4 ), /* x86
Stack size/offset = 4 */
/* 156 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */

/* Parameter txn_out */

/* 158 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
/* 160 */ NdrFcShort( 0x14 ), /* x86
Stack size/offset = 20 */
/* 162 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */

/* Return value */

/* 164 */ NdrFcShort( 0x70 ), /* Flags:
out, return, base type, */
/* 166 */ NdrFcShort( 0x18 ), /* x86
Stack size/offset = 24 */
/* 168 */ 0x8, /*
FC_LONG */
/* 0 */
/* Procedure CallSetComplete */

/* 170 */ 0x33, /*
FC_AUTO_HANDLE */
/* Old Flags: object,
Oi2 */
/* 172 */ NdrFcLong( 0x0 ), /* 0 */
/* 176 */ NdrFcShort( 0x8 ), /* 8 */
/* 178 */ NdrFcShort( 0x8 ), /* x86
Stack size/offset = 8 */
/* 180 */ NdrFcShort( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x8 ), /* 8 */
/* 184 */ 0x4, /* Oi2
Flags: has return, */
/* 1 */
/* Return value */

/* 186 */ NdrFcShort( 0x70 ), /* Flags:
out, return, base type, */
/* 188 */ NdrFcShort( 0x4 ), /* x86
Stack size/offset = 4 */

```

```

/* 190 */ 0x8, /*
FC_LONG */
/* 0 */

}
};

static const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString =
{
0,
{
NdrFcShort( 0x0 ), /* 0 */
/* 2 */
0x12,
0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0x3ca ), /*
Offset= 970 (974) */
/* 6 */
0x2b,
FC_NON_ENCAPSULATED_UNION */
/* 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( 0x2f ), /* 47 */
/* 18 */ NdrFcLong( 0x14 ), /* 20 */
/* 22 */ NdrFcShort( 0x800b ), /*
Simple arm type: FC_HYPER */
/* 24 */ NdrFcLong( 0x3 ), /* 3 */
/* 28 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 30 */ NdrFcLong( 0x11 ), /* 17 */
/* 34 */ NdrFcShort( 0x8001 ), /*
Simple arm type: FC_BYTE */
/* 36 */ NdrFcLong( 0x2 ), /* 2 */
/* 40 */ NdrFcShort( 0x8006 ), /*
Simple arm type: FC_SHORT */
/* 42 */ NdrFcLong( 0x4 ), /* 4 */
/* 46 */ NdrFcShort( 0x800a ), /*
Simple arm type: FC_FLOAT */
/* 48 */ NdrFcLong( 0x5 ), /* 5 */
/* 52 */ NdrFcShort( 0x800c ), /*
Simple arm type: FC_DOUBLE */
/* 54 */ NdrFcLong( 0xb ), /* 11 */
/* 58 */ NdrFcShort( 0x8006 ), /*
Simple arm type: FC_SHORT */
/* 60 */ NdrFcLong( 0xa ), /* 10 */
/* 64 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 66 */ NdrFcLong( 0x6 ), /* 6 */
/* 70 */ NdrFcShort( 0xe8 ), /*
Offset= 232 (302) */
/* 72 */ NdrFcLong( 0x7 ), /* 7 */
/* 76 */ NdrFcShort( 0x800c ), /*
Simple arm type: FC_DOUBLE */
/* 78 */ NdrFcLong( 0x8 ), /* 8 */
/* 82 */ NdrFcShort( 0xe2 ), /*
Offset= 226 (308) */
/* 84 */ NdrFcLong( 0xd ), /* 13 */
/* 88 */ NdrFcShort( 0xf4 ), /*
Offset= 244 (332) */
/* 90 */ NdrFcLong( 0x9 ), /* 9 */

```

```

/* 94 */ NdrFcShort( 0x100 ), /*
Offset= 256 (350) */
/* 96 */ NdrFcLong( 0x2000 ), /* 8192
*/
/* 100 */ NdrFcShort( 0x10c ), /*
Offset= 268 (368) */
/* 102 */ NdrFcLong( 0x24 ), /* 36 */
/* 106 */ NdrFcShort( 0x31a ), /*
Offset= 794 (900) */
/* 108 */ NdrFcLong( 0x4024 ), /* 16420
*/
/* 112 */ NdrFcShort( 0x314 ), /*
Offset= 788 (900) */
/* 114 */ NdrFcLong( 0x4011 ), /* 16401
*/
/* 118 */ NdrFcShort( 0x312 ), /*
Offset= 786 (904) */
/* 120 */ NdrFcLong( 0x4002 ), /* 16386
*/
/* 124 */ NdrFcShort( 0x310 ), /*
Offset= 784 (908) */
/* 126 */ NdrFcLong( 0x4003 ), /* 16387
*/
/* 130 */ NdrFcShort( 0x30e ), /*
Offset= 782 (912) */
/* 132 */ NdrFcLong( 0x4014 ), /* 16404
*/
/* 136 */ NdrFcShort( 0x30c ), /*
Offset= 780 (916) */
/* 138 */ NdrFcLong( 0x4004 ), /* 16388
*/
/* 142 */ NdrFcShort( 0x30a ), /*
Offset= 778 (920) */
/* 144 */ NdrFcLong( 0x4005 ), /* 16389
*/
/* 148 */ NdrFcShort( 0x308 ), /*
Offset= 776 (924) */
/* 150 */ NdrFcLong( 0x400b ), /* 16395
*/
/* 154 */ NdrFcShort( 0x2f2 ), /*
Offset= 754 (908) */
/* 156 */ NdrFcLong( 0x400a ), /* 16394
*/
/* 160 */ NdrFcShort( 0x2f0 ), /*
Offset= 752 (912) */
/* 162 */ NdrFcLong( 0x4006 ), /* 16390
*/
/* 166 */ NdrFcShort( 0x2fa ), /*
Offset= 762 (928) */
/* 168 */ NdrFcLong( 0x4007 ), /* 16391
*/
/* 172 */ NdrFcShort( 0x2f0 ), /*
Offset= 752 (924) */
/* 174 */ NdrFcLong( 0x4008 ), /* 16392
*/
/* 178 */ NdrFcShort( 0x2f2 ), /*
Offset= 754 (932) */
/* 180 */ NdrFcLong( 0x400d ), /* 16397
*/
/* 184 */ NdrFcShort( 0x2f0 ), /*
Offset= 752 (936) */
/* 186 */ NdrFcLong( 0x4009 ), /* 16393
*/
/* 190 */ NdrFcShort( 0x2ee ), /*
Offset= 750 (940) */
/* 192 */ NdrFcLong( 0x6000 ), /* 24576
*/
/* 196 */ NdrFcShort( 0x2ec ), /*
Offset= 748 (944) */
/* 198 */ NdrFcLong( 0x400c ), /* 16396
*/
/* 202 */ NdrFcShort( 0x2ea ), /*
Offset= 746 (948) */
/* 204 */ NdrFcLong( 0x10 ), /* 16 */

```

```

/* 208 */ NdrFcShort( 0x8002 ), /*
Simple arm type: FC_LONG */
/* 216 */ NdrFcLong( 0x12 ), /* 18 */
/* 214 */ NdrFcShort( 0x8006 ), /*
Simple arm type: FC_SHORT */
/* 220 */ NdrFcLong( 0x13 ), /* 19 */
/* 222 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 224 */ NdrFcLong( 0xe ), /* 21 */
/* 226 */ NdrFcShort( 0x800b ), /*
Simple arm type: FC_HYPER */
/* 228 */ NdrFcLong( 0x16 ), /* 22 */
/* 232 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 234 */ NdrFcLong( 0x17 ), /* 23 */
/* 238 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 240 */ NdrFcLong( 0xe ), /* 14 */
/* 244 */ NdrFcShort( 0x2c8 ), /*
Offset= 712 (956) */
/* 246 */ NdrFcLong( 0x400e ), /* 16398
*/
/* 250 */ NdrFcShort( 0x2cc ), /*
Offset= 716 (966) */
/* 252 */ NdrFcLong( 0x4010 ), /* 16400
*/
/* 256 */ NdrFcShort( 0x2ca ), /*
Offset= 714 (970) */
/* 258 */ NdrFcLong( 0x4012 ), /* 16402
*/
/* 262 */ NdrFcShort( 0x286 ), /*
Offset= 646 (908) */
/* 264 */ NdrFcLong( 0x4013 ), /* 16403
*/
/* 268 */ NdrFcShort( 0x284 ), /*
Offset= 644 (912) */
/* 270 */ NdrFcLong( 0x4015 ), /* 16405
*/
/* 274 */ NdrFcShort( 0x282 ), /*
Offset= 642 (916) */
/* 276 */ NdrFcLong( 0x4016 ), /* 16406
*/
/* 280 */ NdrFcShort( 0x278 ), /*
Offset= 632 (912) */
/* 282 */ NdrFcLong( 0x4017 ), /* 16407
*/
/* 286 */ NdrFcShort( 0x272 ), /*
Offset= 626 (912) */
/* 288 */ NdrFcLong( 0x0 ), /* 0 */
/* 292 */ NdrFcShort( 0x0 ), /*
Offset= 0 (292) */
/* 294 */ NdrFcLong( 0x1 ), /* 1 */
/* 298 */ NdrFcShort( 0x0 ), /*
Offset= 0 (298) */
/* 300 */ NdrFcShort( 0xfffffff ), /*
Offset= -1 (299) */
/* 302 */
0x15,
/* FC_STRUCT */
0x7,
/* 7 */
/* 304 */ NdrFcShort( 0x8 ), /* 8 */
/* 306 */ 0xb, /*
FC_HYPER */
0x5b,
/* FC_END */
/* 308 */
0x0, /* FC_UP */
/* 310 */ NdrFcShort( 0xc ), /*
Offset= 12 (322) */
/* 312 */
0x1b,
/* FC_CARRAY */
0x1,
/* 1 */

```

```

/* 316 */ NdrFcShort( 0x2 ), /* 2 */
desc: FC_ULONG */
0x0,
/* */
/* 318 */ NdrFcShort( 0xfffc ), /* -4 */
/* 320 */ 0x6, /*
FC_SHORT */
0x5b,
/* FC_END */
/* 322 */
0x17,
/* FC_CSTRUCT */
0x3,
/* 3 */
/* 324 */ NdrFcShort( 0x8 ), /* 8 */
/* 326 */ NdrFcShort( 0xfffffff2 ), /*
Offset= -14 (312) */
/* 328 */ 0x8, /*
FC_LONG */
0x8,
/* FC_LONG */
/*
FC_PAD */
0x5c,
/* 5c,
FC_PAD */
0x5b,
/* FC_END */
/* 332 */
0x2f,
/* FC_IP */
0x5a,
/* FC_CONSTANT_IID
*/
/* 334 */ NdrFcLong( 0x0 ), /* 0 */
/* 338 */ NdrFcShort( 0x0 ), /* 0 */
/* 340 */ NdrFcShort( 0x0 ), /* 0 */
/* 342 */ 0xc0, /* 192
*/
0x0,
/* 0 */
/* 344 */ 0x0, /* 0 */
/* 0 */
/* 0 */
/* 346 */ 0x0, /* 0 */
0x0,
/* 0 */
/* 348 */ 0x0, /* 0 */
/* 0 */
/* 70 */
/* 350 */
0x2f,
/* FC_IP */
0x5a,
/* FC_CONSTANT_IID
*/
/* 352 */ NdrFcLong( 0x20400 ), /*
132096 */
/* 356 */ NdrFcShort( 0x0 ), /* 0 */
/* 358 */ NdrFcShort( 0x0 ), /* 0 */
/* 360 */ 0xc0, /* 192
*/
0x0,
/* 0 */
/* 362 */ 0x0, /* 0 */
0x0,
/* 0 */
/* 364 */ 0x0, /* 0 */
/* 0 */
/* 366 */ 0x0, /* 0 */
0x46,
/* 70 */
/* 368 */
0x12,
/* FC_UP [pointer_deref] */
0x10,

```

```

/* 370 */ NdrFcShort( 0x2 ), /*
Offset= 2 (372) */
/* 372 */
0x0, /* FC_UP */
/* 374 */ NdrFcShort( 0x1fc ), /*
Offset= 508 (882) */
/* 376 */
/*
FC_ENCAPSULATED_UNION */
/* 73 */
/* 378 */ NdrFcShort( 0x18 ), /* 24 */
/* 380 */ NdrFcShort( 0xa ), /* 10 */
/* 382 */ NdrFcLong( 0x8 ), /* 8 */
/* 386 */ NdrFcShort( 0x58 ), /*
Offset= 88 (474) */
/* 388 */ NdrFcLong( 0xd ), /* 13 */
/* 392 */ NdrFcShort( 0x78 ), /*
Offset= 120 (512) */
/* 394 */ NdrFcLong( 0x9 ), /* 9 */
/* 398 */ NdrFcShort( 0x94 ), /*
Offset= 148 (546) */
/* 400 */ NdrFcLong( 0xc ), /* 12 */
/* 404 */ NdrFcShort( 0xbc ), /*
Offset= 188 (592) */
/* 406 */ NdrFcLong( 0x24 ), /* 36 */
/* 410 */ NdrFcShort( 0x114 ), /*
Offset= 276 (686) */
/* 412 */ NdrFcLong( 0x800d ), /* 32781
*/
/* 416 */ NdrFcShort( 0x130 ), /*
Offset= 304 (720) */
/* 418 */ NdrFcLong( 0x10 ), /* 16 */
/* 422 */ NdrFcShort( 0x148 ), /*
Offset= 328 (750) */
/* 424 */ NdrFcLong( 0x2 ), /* 2 */
/* 428 */ NdrFcShort( 0x160 ), /*
Offset= 352 (780) */
/* 430 */ NdrFcLong( 0x3 ), /* 3 */
/* 434 */ NdrFcShort( 0x178 ), /*
Offset= 376 (810) */
/* 436 */ NdrFcLong( 0x14 ), /* 20 */
/* 440 */ NdrFcShort( 0x190 ), /*
Offset= 400 (840) */
/* 442 */ NdrFcShort( 0xffffffff ), /*
Offset= -1 (441) */
/* 444 */
0x1b, /* FC_CARRAY */
0x3, /* 3 */
/* 446 */ NdrFcShort( 0x4 ), /* 4 */
/* 448 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
0x0, /*
*/
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 454 */
0x48, /*
FC_VARIABLE_REPEAT */
0x49, /* FC_FIXED_OFFSET
*/
/* 456 */ NdrFcShort( 0x4 ), /* 4 */
/* 458 */ NdrFcShort( 0x0 ), /* 0 */
/* 460 */ NdrFcShort( 0x1 ), /* 1 */
/* 462 */ NdrFcShort( 0x0 ), /* 0 */

```

```

/* 466 */ NdrFcShort( 0x0fc ), /* 0 */
/* 468 */ NdrFcShort( 0xffffffff6e ), /*
Offset= -146 (322) */
/* 470 */
0x5b, /* FC_END */
0x8, /* FC_LONG */
/* 472 */ 0x5c, /*
FC_PAD */
/* 474 */
0x5b, /* FC_END */
0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 476 */ NdrFcShort( 0x8 ), /* 8 */
/* 478 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 480 */
0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 482 */ NdrFcShort( 0x4 ), /* 4 */
/* 484 */ NdrFcShort( 0x4 ), /* 4 */
/* 486 */ 0x11, 0x0, /* FC_RP */
/* 488 */ NdrFcShort( 0xffffffffd4 ), /*
Offset= -44 (444) */
/* 490 */
0x5b, /* FC_END */
0x8, /* FC_LONG */
/* 492 */ 0x8, /*
FC_LONG */
0x5b, /* FC_END */
/* 494 */
0x21, /* FC_BOGUS_ARRAY
*/
0x3, /* 3 */
/* 496 */ NdrFcShort( 0x0 ), /* 0 */
/* 498 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
0x0, /*
*/
/* 500 */ NdrFcShort( 0x0 ), /* 0 */
/* 502 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 506 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 508 */ NdrFcShort( 0xffffffff50 ), /*
Offset= -176 (332) */
/* 510 */ 0x5c, /*
FC_PAD */
/* 512 */
0x5b, /* FC_END */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 514 */ NdrFcShort( 0x8 ), /* 8 */
/* 516 */ NdrFcShort( 0x0 ), /* 0 */

```

```

/* 518 */ NdrFcShort( 0x6 ), /*
Offset= 6 (524) */
/* 520 */
0x36, /* FC_POINTER */
0x5c, /*
FC_PAD */
/* 524 */
0x5b, /* FC_END */
0x11, /*
0x0, /* FC_RP */
/* 526 */ NdrFcShort( 0xffffffffe0 ), /*
Offset= -32 (494) */
/* 528 */
0x21, /* FC_BOGUS_ARRAY
*/
0x3, /* 3 */
/* 530 */ NdrFcShort( 0x0 ), /* 0 */
/* 532 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
0x0, /*
*/
/* 534 */ NdrFcShort( 0x0 ), /* 0 */
/* 536 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 540 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 542 */ NdrFcShort( 0xffffffff40 ), /*
Offset= -192 (350) */
/* 544 */ 0x5c, /*
FC_PAD */
0x5b, /* FC_END */
/* 546 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 548 */ NdrFcShort( 0x8 ), /* 8 */
/* 550 */ NdrFcShort( 0x0 ), /* 0 */
/* 552 */ NdrFcShort( 0x6 ), /*
Offset= 6 (558) */
/* 554 */ 0x8, /*
FC_LONG */
0x36, /* FC_POINTER */
0x5c, /*
FC_PAD */
0x5b, /* FC_END */
/* 558 */
0x11, /*
0x0, /* FC_RP */
/* 560 */ NdrFcShort( 0xffffffffe0 ), /*
Offset= -32 (528) */
/* 562 */
0x1b, /* FC_CARRAY */
0x3, /* 3 */
/* 564 */ NdrFcShort( 0x4 ), /* 4 */
/* 566 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
0x0, /*
*/
/* 568 */ NdrFcShort( 0x0 ), /* 0 */
/* 570 */

```

```

                                0x4b,
/* FC_PP */
                                0x5c,
/* FC_PAD */
/* 572 */
                                0x48,
FC_VARIABLE_REPEAT */
                                0x49,
/* FC_FIXED_OFFSET
*/
/* 574 */ NdrFcShort( 0x4 ), /* 4 */
/* 576 */ NdrFcShort( 0x0 ), /* 0 */
/* 578 */ NdrFcShort( 0x1 ), /* 1 */
/* 580 */ NdrFcShort( 0x0 ), /* 0 */
/* 582 */ NdrFcShort( 0x0 ), /* 0 */
/* 584 */ 0x12, 0x0, /* FC_UP */
/* 586 */ NdrFcShort( 0x184 ), /*
Offset= 388 (974) */
/* 588 */
                                0x5b,
/* FC_END */
                                0x8,
/* FC_LONG */
/* 590 */ 0x5c, /*
FC_PAD */
                                0x5b,
/* FC_END */
/* 592 */
                                0x1a,
FC_BOGUS_STRUCT */
                                3,
/* 3 */
/* 594 */ NdrFcShort( 0x8 ), /* 8 */
/* 596 */ NdrFcShort( 0x0 ), /* 0 */
/* 598 */ NdrFcShort( 0x6 ), /*
Offset= 6 (604) */
/* 600 */ 0x8, /*
FC_LONG */
                                36,
/* FC_POINTER */
/* 602 */ 0x5c, /*
FC_PAD */
                                5b,
/* FC_END */
/* 604 */
                                11,
0x0, /* FC_RP */
/* 606 */ NdrFcShort( 0xffffd4 ), /*
Offset= -44 (562) */
/* 608 */
                                2f,
/* FC_IP */
                                5a,
/* FC_CONSTANT_IID
*/
/* 610 */ NdrFcLong( 0x2f ), /* 47 */
/* 614 */ NdrFcShort( 0x0 ), /* 0 */
/* 616 */ NdrFcShort( 0x0 ), /* 0 */
/* 618 */ 0xc0, /* 192
*/
                                0,
/* 620 */ 0x0, /* 0 */
                                0,
/* 622 */ 0x0, /* 0 */
                                0,
/* 624 */ 0x0, /* 0 */
                                46,
/* 626 */
                                70 */

```

```

                                0x1b,
/* FC_CARRAY */
/* 628 */ NdrFcShort( 0x1 ), /* 1 */
/* 630 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
                                0,
/* 632 */ NdrFcShort( 0x4 ), /* 4 */
/* 634 */ 0x1, /*
FC_BYTE */
                                5b,
/* FC_END */
/* 636 */
FC_BOGUS_STRUCT */
                                3,
/* 3 */
/* 638 */ NdrFcShort( 0x10 ), /* 16 */
/* 640 */ NdrFcShort( 0x0 ), /* 0 */
/* 642 */ NdrFcShort( 0xa ), /*
Offset= 10 (652) */
/* 644 */ 0x8, /*
FC_LONG */
                                8,
/* FC_LONG */
/* 646 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
                                0,
/* 648 */ NdrFcShort( 0xfffffd8 ), /*
Offset= -40 (608) */
/* 650 */ 0x36, /*
FC_POINTER */
                                5b,
/* FC_END */
/* 652 */
                                12,
0x0, /* FC_UP */
/* 654 */ NdrFcShort( 0xfffffe4 ), /*
Offset= -28 (626) */
/* 656 */
                                1b,
/* FC_CARRAY */
                                3,
/* 3 */
/* 658 */ NdrFcShort( 0x4 ), /* 4 */
/* 660 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
                                0,
/* 662 */ NdrFcShort( 0x0 ), /* 0 */
/* 664 */
                                4b,
/* FC_PP */
                                5c,
/* FC_PAD */
                                48,
FC_VARIABLE_REPEAT */
                                49,
/* FC_FIXED_OFFSET
*/
/* 668 */ NdrFcShort( 0x4 ), /* 4 */
/* 670 */ NdrFcShort( 0x0 ), /* 0 */
/* 672 */ NdrFcShort( 0x1 ), /* 1 */
/* 674 */ NdrFcShort( 0x0 ), /* 0 */
/* 676 */ NdrFcShort( 0x0 ), /* 0 */
/* 678 */ 0x12, 0x0, /* FC_UP */
/* 680 */ NdrFcShort( 0xfffffd4 ), /*
Offset= -44 (636) */
/* 682 */

```

```

                                5b,
/* FC_END */
                                8,
/* FC_LONG */
/* 684 */ 0x5c, /*
FC_PAD */
                                5b,
/* FC_END */
                                1a,
FC_BOGUS_STRUCT */
                                3,
/* 3 */
/* 688 */ NdrFcShort( 0x8 ), /* 8 */
/* 690 */ NdrFcShort( 0x0 ), /* 0 */
/* 692 */ NdrFcShort( 0x6 ), /*
Offset= 6 (698) */
/* 694 */ 0x8, /*
FC_LONG */
                                36,
/* FC_POINTER */
/* 696 */ 0x5c, /*
FC_PAD */
                                5b,
/* FC_END */
/* 698 */
                                11,
0x0, /* FC_RP */
/* 700 */ NdrFcShort( 0xfffffd4 ), /*
Offset= -44 (656) */
/* 702 */
                                1d,
/* FC_SMFARRAY */
                                0,
/* 704 */ NdrFcShort( 0x8 ), /* 8 */
/* 706 */ 0x1, /*
FC_BYTE */
                                5b,
/* FC_END */
/* 708 */
                                15,
/* FC_STRUCT */
                                3,
/* 3 */
/* 710 */ NdrFcShort( 0x10 ), /* 16 */
/* 712 */ 0x8, /*
FC_LONG */
                                6,
/* FC_SHORT */
/* 714 */ 0x6, /*
FC_SHORT */
                                4c,
/* FC_EMBEDDED_COMPLEX */
/* 716 */ 0x0, /* 0 */
                                1,
NdrFcShort( 0xfffff1 ), /*
Offset= -15 (702) */
                                5b,
/* FC_END */
/* 720 */
                                1a,
FC_BOGUS_STRUCT */
                                3,
/* 3 */
/* 722 */ NdrFcShort( 0x18 ), /* 24 */
/* 724 */ NdrFcShort( 0x0 ), /* 0 */
/* 726 */ NdrFcShort( 0xa ), /*
Offset= 10 (736) */

```

```

/* 728 */ 0x8, /*
FC_LONG */
/* 730 */ 0x4c, /* FC_POINTER */
FC_EMBEDDED_COMPLEX */
/* 732 */ NdrFcShort( 0xfffffe8 ), /*
Offset=-24 (708) */
/* 734 */ 0x5c, /*
FC_PAD */
/* 736 */
0x0, /* FC_RP */
/* 738 */ NdrFcShort( 0xfffff0c ), /*
Offset=-244 (494) */
/* 740 */
/* 742 */ NdrFcShort( 0x1 ), /* 1 */
/* 744 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
/* 746 */ NdrFcShort( 0x0 ), /* 0 */
/* 748 */ 0x1, /*
FC_BYTE */
/* 750 */
/* 752 */ NdrFcShort( 0x8 ), /* 8 */
/* 754 */
/* 756 */
/* 758 */ NdrFcShort( 0x4 ), /* 4 */
/* 760 */ NdrFcShort( 0x4 ), /* 4 */
/* 762 */ 0x12, 0x0, /* FC_UP */
/* 764 */ NdrFcShort( 0xfffffe8 ), /*
Offset=-24 (740) */
/* 766 */
/* 768 */ 0x8, /*
FC_LONG */
/* 770 */
/* 772 */ NdrFcShort( 0x2 ), /* 2 */
/* 774 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */

```

```

/* 776 */ NdrFcShort( 0x0 ), /* 0 */
FC_SHORT */
/* 780 */
/* 782 */ NdrFcShort( 0x8 ), /* 8 */
/* 784 */
/* 786 */
/* 788 */ NdrFcShort( 0x4 ), /* 4 */
/* 790 */ NdrFcShort( 0x4 ), /* 4 */
/* 792 */ 0x12, 0x0, /* FC_UP */
/* 794 */ NdrFcShort( 0xfffffe8 ), /*
Offset=-24 (770) */
/* 796 */
/* 798 */ 0x8, /*
FC_LONG */
/* 800 */
/* 802 */ NdrFcShort( 0x4 ), /* 4 */
/* 804 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
/* 806 */ NdrFcShort( 0x0 ), /* 0 */
/* 808 */ 0x8, /*
FC_LONG */
/* 810 */
/* 812 */ NdrFcShort( 0x8 ), /* 8 */
/* 814 */
/* 816 */
/* 818 */ NdrFcShort( 0x4 ), /* 4 */
/* 820 */ NdrFcShort( 0x4 ), /* 4 */
/* 822 */ 0x12, 0x0, /* FC_UP */
/* 824 */ NdrFcShort( 0xfffffe8 ), /*
Offset=-24 (800) */
/* 826 */

```

```

/* 828 */ 0x8, /* FC_LONG */
/* 830 */
/* 832 */ NdrFcShort( 0x8 ), /* 8 */
/* 834 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
/* 836 */ NdrFcShort( 0x0 ), /* 0 */
/* 838 */ 0xb, /*
FC_HYPER */
/* 840 */
/* 842 */ NdrFcShort( 0x8 ), /* 8 */
/* 844 */
/* 846 */
/* 848 */ NdrFcShort( 0x4 ), /* 4 */
/* 850 */ NdrFcShort( 0x4 ), /* 4 */
/* 852 */ 0x12, 0x0, /* FC_UP */
/* 854 */ NdrFcShort( 0xfffffe8 ), /*
Offset=-24 (830) */
/* 856 */
/* 858 */ 0x8, /*
FC_LONG */
/* 860 */
/* 862 */ NdrFcShort( 0x8 ), /* 8 */
/* 864 */ 0x8, /*
FC_LONG */
/* 866 */ 0x5c, /*
FC_PAD */
/* 868 */
/* 870 */ NdrFcShort( 0x8 ), /* 8 */
/* 872 */ 0x7, /* Corr
desc: FC_USHORT */

```

```

/* */ 0x0,
/* 874 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 876 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 0 */ 0x0,
/* 878 */ NdrFcShort( 0xfffffee ), /*
Offset=-18 (860) */
/* 880 */ 0x5c, /*
FC_PAD */
/* 882 */ /* FC_END */
/* 0x5b,
0x1a,
FC_BOGUS_STRUCT */
/* 0x3,
/* 3 */
/* 884 */ NdrFcShort( 0x28 ), /* 40 */
/* 886 */ NdrFcShort( 0xfffffee ), /*
Offset=-18 (868) */
/* 888 */ NdrFcShort( 0x0 ), /*
Offset=0 (888) */
/* 890 */ 0x6, /*
FC_SHORT */
/* 892 */ 0x8, /* FC_SHORT */
/* 894 */ 0x4c, /* FC_LONG */
/* 896 */ NdrFcShort( 0xffffdf8 ), /*
Offset=-520 (376) */
/* 898 */ 0x5c, /*
FC_PAD */
/* 900 */ /* FC_END */
0x0, /* FC_UP */
/* 902 */ NdrFcShort( 0xfffffef ), /*
Offset=-266 (636) */
/* 904 */
0x8, /* FC_UP [simple_pointer] */
/* 906 */ 0x1, /*
FC_BYTE */
/* 908 */ /* FC_PAD */
0x8, /* FC_UP [simple_pointer] */
/* 910 */ 0x6, /*
FC_SHORT */
/* 912 */ /* FC_PAD */
0x8, /* FC_UP [simple_pointer] */
/* 914 */ 0x8, /*
FC_LONG */
/* 916 */ /* FC_PAD */
0x8, /* FC_UP [simple_pointer] */
/* 918 */ 0xb, /*
FC_HYPER */
/* 920 */ 0x5c, /*
FC_PAD */

```

```

/* 920 */ 0x12,
0x8, /* FC_UP [simple_pointer] */
/* 922 */ 0xa, /*
FC_FLOAT */
/* 924 */ /* FC_PAD */
0x8, /* FC_UP [simple_pointer] */
/* 926 */ 0xc, /*
FC_DOUBLE */
/* 928 */ /* FC_PAD */
0x0, /* FC_UP */
/* 930 */ NdrFcShort( 0xffffd8c ), /*
Offset=-628 (302) */
/* 932 */
0x10, /* FC_UP [pointer_deref] */
/* 934 */ NdrFcShort( 0xffffd8e ), /*
Offset=-626 (308) */
/* 936 */
0x10, /* FC_UP [pointer_deref] */
/* 938 */ NdrFcShort( 0xffffda2 ), /*
Offset=-606 (332) */
/* 940 */
0x10, /* FC_UP [pointer_deref] */
/* 942 */ NdrFcShort( 0xffffdb0 ), /*
Offset=-592 (350) */
/* 944 */
0x10, /* FC_UP [pointer_deref] */
/* 946 */ NdrFcShort( 0xffffdbe ), /*
Offset=-578 (368) */
/* 948 */
0x10, /* FC_UP [pointer_deref] */
/* 950 */ NdrFcShort( 0x2 ), /*
Offset=2 (952) */
/* 952 */
0x0, /* FC_UP */
/* 954 */ NdrFcShort( 0x14 ), /*
Offset=20 (974) */
/* 956 */
0x15, /* FC_STRUCT */
0x7, /* 7 */
/* 958 */ NdrFcShort( 0x10 ), /* 16 */
/* 960 */ 0x6, /*
FC_SHORT */
/* 962 */ 0x1, /* FC_BYTE */
/* 964 */ 0xb, /* FC_LONG */
/* 966 */ /* FC_HYPER */
/* 968 */ 0x5b, /* FC_END */
/* 970 */
0x8, /* FC_UP [simple_pointer] */

```

```

/* 972 */ 0x2, /*
FC_CHAR */
/* 974 */ /* FC_PAD */
/* 976 */ NdrFcShort( 0x20 ), /* 7 */
/* 978 */ NdrFcShort( 0x0 ), /* 32 */
/* 980 */ NdrFcShort( 0x0 ), /* 0 */
Offset=0 (980)
/* 982 */ 0x8, /*
FC_LONG */
/* 984 */ 0x6, /* FC_SHORT */
/* 986 */ 0x6, /* FC_SHORT */
/* 988 */ 0x4c, /* FC_SHORT */
/* 990 */ NdrFcShort( 0xffffc28 ), /*
Offset=-984 (6) */
/* 992 */ 0x5c, /*
FC_PAD */
/* 994 */ 0xb4, /* FC_END */
/* 996 */ NdrFcShort( 0x0 ), /* 0 */
/* 998 */ NdrFcShort( 0x10 ), /* 16 */
/* 1000 */ NdrFcShort( 0x0 ), /* 0 */
/* 1002 */ NdrFcShort( 0xffffc18 ), /*
Offset=-1000 (2) */
/* 1004 */
0x4, /* FC_UP */
/* 1006 */ NdrFcShort( 0x6 ), /*
Offset=6 (1012) */
/* 1008 */
0x0, /* FC_OP */
/* 1010 */ NdrFcShort( 0xffffcdc ), /*
Offset=-36 (974) */
/* 1012 */ 0xb4, /*
FC_USER_MARSHAL */
/* 1014 */ NdrFcShort( 0x0 ), /* 131 */
/* 1016 */ NdrFcShort( 0x10 ), /* 0 */
/* 1018 */ NdrFcShort( 0x0 ), /* 16 */
/* 1020 */ NdrFcShort( 0xfffff4 ), /* 0 */
Offset=-12 (1008)
}
};
static const
USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ] =
{

```

```

    {
        VARIANT_UserSize
        ,VARIANT_UserMarshal
        ,VARIANT_UserUnmarshal
        ,VARIANT_UserFree
    }
};

/* Standard interface:
__MIDL_if_tpc_com_ps_0000, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00}} */

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

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

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

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

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

CINTERFACE_PROXY_VTABLE(9)
_ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    (void *) (INT_PTR) -1 /* ITPCC::NewOrder */
};

```

```

(void *) (INT_PTR) -1 /* ITPCC::NewOrder */,,
(void *) (INT_PTR) -1 /* ITPCC::StockLevel */
/
(void *) (INT_PTR) -1 /* ITPCC::OrderStatus
*/ ,
(void *) (INT_PTR) -1 /*
ITPCC::CallSetComplete */
*/;

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,
    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

static const MIDL_STUB_DESC Object_StubDesc
=
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x600015b, /* MIDL Version 6.0.347 */
    0,
    UserMarshalRoutines,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* cs routines */
    0, /* proxy/server info */
    0 /* Reserved5 */
};

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

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

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

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

int __stdcall _tpcc_com_ps_IID_Lookup( const
IID * pIID, int * pIndex )
{
    if(!_tpcc_com_ps_CHECK_IID(0))
    {

```

```

return 0;
}

const ExtendedProxyFileInfo
tpcc_com_ps_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) &
_tpc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) &
_tpc_com_ps_StubVtblList,
    (const PCInterfaceName *) &
_tpc_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 */
};

#ifdef !defined(_M_IA64) &&
defined(_M_AMD64)*/

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

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

/* File created by MIDL compiler version
6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf, W1, Zp8, env=Win64 (32b
run,appending)
protocol : dce , 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_HEADERING( )

#ifdef defined(_M_IA64) || defined(_M_AMD64)
#define USE_STUBLESS_PROXY

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

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

```

```

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 1003
#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;

static RPC_SYNTAX_IDENTIFIER
_RpcTransferSyntax =
{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,
0x00,0x2B,0x10,0x48,0x60}},{2,0}};

extern const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING
__MIDL_ProcFormatString;

extern const MIDL_STUB_DESC
Object_StubDesc;

extern const MIDL_SERVER_INFO
ITPCC_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO
ITPCC_ProxyInfo;

extern const
USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ];

#if !defined(_RPC_WIN64_)
#error Invalid build platform for this stub.
#endif

static const MIDL_PROC_FORMAT_STRING
__MIDL_ProcFormatString =
{
    0,
    {
        /* Procedure NewOrder */
        0x33,
        /* FC_AUTO_HANDLE */
        0x6C,
        /* Old Flags: object,
Oi2 */
        /* 2 */ NdrFcLong( 0x0 ), /* 0 */
        /* 6 */ NdrFcShort( 0x3 ), /* 3 */
        /* 8 */ NdrFcShort( 0x30 ), /* ia64
Stack size/offset = 48 */
        /* 10 */ NdrFcShort( 0x0 ), /* 0 */
        /* 12 */ NdrFcShort( 0x8 ), /* 8 */
    }
}

```

```

/* 14 */ 0x47, /* Oi2
Flags: srv must size, clt must size, has return,
has ext, */
/* 16 */ 0xa, /* 10 */
/* 18 */ NdrFcShort( 0x20 ), /* 32 */
/* 20 */ NdrFcShort( 0x20 ), /* 32 */
/* 22 */ NdrFcShort( 0x0 ), /* 0 */
/* 24 */ NdrFcShort( 0x0 ), /* 0 */
/* Ext Flags: new
corr desc, clt corr check, srv corr check, */
/* Parameter txn_in */
/* 26 */ NdrFcShort( 0x8b ), /* Flags:
must size, must free, in, by val, */
/* 28 */ NdrFcShort( 0x8 ), /* ia64
Stack size/offset = 8 */
/* 30 */ NdrFcShort( 0x3ce ), /* Type
Offset=974 */
/* Parameter txn_out */
/* 32 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
/* 34 */ NdrFcShort( 0x20 ), /* ia64
Stack size/offset = 32 */
/* 36 */ NdrFcShort( 0x3e0 ), /* Type
Offset=992 */
/* Return value */
/* 38 */ NdrFcShort( 0x70 ), /* Flags:
out, return, base type, */
/* 40 */ NdrFcShort( 0x28 ), /* ia64
Stack size/offset = 40 */
/* 42 */ 0x8, /*
FC_LONG */
/* 0 */
/* Procedure Payment */
/* 44 */ 0x33, /*
FC_AUTO_HANDLE */
/* Old Flags: object,
Oi2 */
/* 46 */ NdrFcLong( 0x0 ), /* 0 */
/* 50 */ NdrFcShort( 0x4 ), /* 4 */
/* 52 */ NdrFcShort( 0x30 ), /* ia64
Stack size/offset = 48 */
/* 54 */ NdrFcShort( 0x0 ), /* 0 */
/* 56 */ NdrFcShort( 0x8 ), /* 8 */
/* 58 */ 0x47, /* Oi2
Flags: srv must size, clt must size, has return,
has ext, */
/* 60 */ 0xa, /* 10 */
/* Ext Flags: new
corr desc, clt corr check, srv corr check, */
/* 62 */ NdrFcShort( 0x20 ), /* 32 */
/* 64 */ NdrFcShort( 0x20 ), /* 32 */
/* 66 */ NdrFcShort( 0x0 ), /* 0 */
/* 68 */ NdrFcShort( 0x0 ), /* 0 */
/* Parameter txn_in */
/* 70 */ NdrFcShort( 0x8b ), /* Flags:
must size, must free, in, by val, */

```

```

/* 72 */ NdrFcShort( 0x8 ), /* ia64
Stack size/offset = 8 */
/* 74 */ NdrFcShort( 0x3ce ), /* Type
Offset=974 */
/* Parameter txn_out */
/* 76 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
/* 78 */ NdrFcShort( 0x20 ), /* ia64
Stack size/offset = 32 */
/* 80 */ NdrFcShort( 0x3e0 ), /* Type
Offset=992 */
/* Return value */
/* 82 */ NdrFcShort( 0x70 ), /* Flags:
out, return, base type, */
/* 84 */ NdrFcShort( 0x28 ), /* ia64
Stack size/offset = 40 */
/* 86 */ 0x8, /*
FC_LONG */
/* 0 */
/* Procedure Delivery */
/* 88 */ 0x33, /*
FC_AUTO_HANDLE */
/* Old Flags: object,
Oi2 */
/* 90 */ NdrFcLong( 0x0 ), /* 0 */
/* 94 */ NdrFcShort( 0x5 ), /* 5 */
/* 96 */ NdrFcShort( 0x30 ), /* ia64
Stack size/offset = 48 */
/* 98 */ NdrFcShort( 0x0 ), /* 0 */
/* 100 */ NdrFcShort( 0x8 ), /* 8 */
/* 102 */ 0x47, /* Oi2
Flags: srv must size, clt must size, has return,
has ext, */
/* 0 */
/* 104 */ 0xa, /* 10 */
/* 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, */
/* 116 */ NdrFcShort( 0x8 ), /* ia64
Stack size/offset = 8 */
/* 118 */ NdrFcShort( 0x3ce ), /* Type
Offset=974 */
/* Parameter txn_out */
/* 120 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
/* 122 */ NdrFcShort( 0x20 ), /* ia64
Stack size/offset = 32 */
/* 124 */ NdrFcShort( 0x3e0 ), /* Type
Offset=992 */
/* Return value */

```



```

/* 126 */ NdrFcShort( 0x70 ), /* Flags:
out, return, base type, */
/* 128 */ NdrFcShort( 0x28 ), /* ia64
Stack size/offset = 40 */
/* 130 */ 0x8, /*
FC_LONG */
0x0,
/* 0 */
/* Procedure StockLevel */
/* 132 */ 0x33, /*
FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */
/* 134 */ NdrFcLong( 0x0 ), /* 0 */
/* 138 */ NdrFcShort( 0x6 ), /* 6 */
/* 140 */ NdrFcShort( 0x30 ), /* ia64
Stack size/offset = 48 */
/* 142 */ NdrFcShort( 0x0 ), /* 0 */
/* 144 */ NdrFcShort( 0x8 ), /* 8 */
/* 146 */ 0x47, /* Oi2
Flags: srv must size, clt must size, has return,
has ext, */
0x3,
/* 3 */
/* 148 */ 0xa, /* 10 */
0x7, /* Ext Flags: new
corr desc, clt corr check, srv corr check, */
/* 150 */ NdrFcShort( 0x20 ), /* 32 */
/* 152 */ NdrFcShort( 0x20 ), /* 32 */
/* 154 */ NdrFcShort( 0x0 ), /* 0 */
/* 156 */ NdrFcShort( 0x0 ), /* 0 */
/* Parameter txn_in */
/* 158 */ NdrFcShort( 0x8b ), /* Flags:
must size, must free, in, by val, */
/* 160 */ NdrFcShort( 0x8 ), /* ia64
Stack size/offset = 8 */
/* 162 */ NdrFcShort( 0x3ce ), /* Type
Offset=974 */
/* Parameter txn_out */
/* 164 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
/* 166 */ NdrFcShort( 0x20 ), /* ia64
Stack size/offset = 32 */
/* 168 */ NdrFcShort( 0x3e0 ), /* Type
Offset=992 */
/* Return value */
/* 170 */ NdrFcShort( 0x70 ), /* Flags:
out, return, base type, */
/* 172 */ NdrFcShort( 0x28 ), /* ia64
Stack size/offset = 40 */
/* 174 */ 0x8, /*
FC_LONG */
0x0,
/* 0 */
/* Procedure OrderStatus */
/* 176 */ 0x33, /*
FC_AUTO_HANDLE */
0x6c, /* Old Flags: object,
Oi2 */
/* 178 */ NdrFcLong( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x7 ), /* 7 */

```

```

/* 184 */ NdrFcShort( 0x30 ), /* ia64
Stack size/offset = 48 */
/* 188 */ NdrFcShort( 0x8 ), /* 8 */
/* 190 */ 0x47, /* Oi2
Flags: srv must size, clt must size, has return,
has ext, */
0x3,
/* 3 */
/* 192 */ 0xa, /* 10 */
0x7, /* Ext Flags: new
corr desc, clt corr check, srv corr check, */
/* 194 */ NdrFcShort( 0x20 ), /* 32 */
/* 196 */ NdrFcShort( 0x20 ), /* 32 */
/* 198 */ NdrFcShort( 0x0 ), /* 0 */
/* 200 */ NdrFcShort( 0x0 ), /* 0 */
/* Parameter txn_in */
/* 202 */ NdrFcShort( 0x8b ), /* Flags:
must size, must free, in, by val, */
/* 204 */ NdrFcShort( 0x8 ), /* ia64
Stack size/offset = 8 */
/* 206 */ NdrFcShort( 0x3ce ), /* Type
Offset=974 */
/* Parameter txn_out */
/* 208 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
/* 210 */ NdrFcShort( 0x20 ), /* ia64
Stack size/offset = 32 */
/* 212 */ NdrFcShort( 0x3e0 ), /* Type
Offset=992 */
/* Return value */
/* 214 */ NdrFcShort( 0x70 ), /* Flags:
out, return, base type, */
/* 216 */ NdrFcShort( 0x28 ), /* ia64
Stack size/offset = 40 */
/* 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
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
Stack size/offset = 8 */
/* 250 */ 0x8, /*
FC_LONG */
0x0,
/* 0 */
}
};
static const MIDL_TYPE_FORMAT_STRING
_MIDL_TypeFormatString =
{
0,
{
NdrFcShort( 0x0 ), /* 0 */
0x12,
0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0x3b6 ), /*
Offset= 950 (954) */
/* 6 */
0x2b,
/*
FC_NON_ENCAPSULATED_UNION */
0x9, /* FC_ULONGLONG */
/* 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( 0x2f ), /* 47 */
/* 20 */ NdrFcLong( 0x14 ), /* 20 */
/* 24 */ NdrFcShort( 0x800b ), /*
Simple arm type: FC_HYPER */
/* 26 */ NdrFcLong( 0x3 ), /* 3 */
/* 30 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 32 */ NdrFcLong( 0x11 ), /* 17 */
/* 36 */ NdrFcShort( 0x8001 ), /*
Simple arm type: FC_BYTE */
/* 38 */ NdrFcLong( 0x2 ), /* 2 */
/* 42 */ NdrFcShort( 0x8006 ), /*
Simple arm type: FC_SHORT */
/* 44 */ NdrFcLong( 0x4 ), /* 4 */
/* 48 */ NdrFcShort( 0x800a ), /*
Simple arm type: FC_FLOAT */
/* 50 */ NdrFcLong( 0x5 ), /* 5 */
/* 54 */ NdrFcShort( 0x800c ), /*
Simple arm type: FC_DOUBLE */
/* 56 */ NdrFcLong( 0xb ), /* 11 */
/* 60 */ NdrFcShort( 0x8006 ), /*
Simple arm type: FC_SHORT */
/* 62 */ NdrFcLong( 0xa ), /* 10 */
/* 66 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 68 */ NdrFcLong( 0x6 ), /* 6 */
/* 72 */ NdrFcShort( 0xe8 ), /*
Offset= 232 (304) */
/* 74 */ NdrFcLong( 0x7 ), /* 7 */
/* 78 */ NdrFcShort( 0x800c ), /*
Simple arm type: FC_DOUBLE */
/* 80 */ NdrFcLong( 0x8 ), /* 8 */

```

```

/* 84 */ NdrFcShort( 0xe2 ), /*
Offset= 226 (310) */
/* 86 */ NdrFcLong( 0xd ), /* 13 */
/* 90 */ NdrFcShort( 0xf6 ), /*
Offset= 246 (336) */
/* 92 */ NdrFcLong( 0x9 ), /* 9 */
/* 96 */ NdrFcShort( 0x102 ), /*
Offset= 258 (354) */
/* 98 */ NdrFcLong( 0x2000 ), /* 8192
*/
/* 102 */ NdrFcShort( 0x10e ), /*
Offset= 270 (372) */
/* 104 */ NdrFcLong( 0x24 ), /* 36 */
/* 108 */ NdrFcShort( 0x304 ), /*
Offset= 772 (880) */
/* 110 */ NdrFcLong( 0x4024 ), /* 16420
*/
/* 114 */ NdrFcShort( 0x2fe ), /*
Offset= 766 (880) */
/* 116 */ NdrFcLong( 0x4011 ), /* 16401
*/
/* 120 */ NdrFcShort( 0x2fc ), /*
Offset= 764 (884) */
/* 122 */ NdrFcLong( 0x4002 ), /* 16386
*/
/* 126 */ NdrFcShort( 0x2fa ), /*
Offset= 762 (888) */
/* 128 */ NdrFcLong( 0x4003 ), /* 16387
*/
/* 132 */ NdrFcShort( 0x2f8 ), /*
Offset= 760 (892) */
/* 134 */ NdrFcLong( 0x4014 ), /* 16404
*/
/* 138 */ NdrFcShort( 0x2f6 ), /*
Offset= 758 (896) */
/* 140 */ NdrFcLong( 0x4004 ), /* 16388
*/
/* 144 */ NdrFcShort( 0x2f4 ), /*
Offset= 756 (900) */
/* 146 */ NdrFcLong( 0x4005 ), /* 16389
*/
/* 150 */ NdrFcShort( 0x2f2 ), /*
Offset= 754 (904) */
/* 152 */ NdrFcLong( 0x400b ), /* 16395
*/
/* 156 */ NdrFcShort( 0x2dc ), /*
Offset= 732 (888) */
/* 158 */ NdrFcLong( 0x400a ), /* 16394
*/
/* 162 */ NdrFcShort( 0x2da ), /*
Offset= 730 (892) */
/* 164 */ NdrFcLong( 0x4006 ), /* 16390
*/
/* 168 */ NdrFcShort( 0x2e4 ), /*
Offset= 740 (908) */
/* 170 */ NdrFcLong( 0x4007 ), /* 16391
*/
/* 174 */ NdrFcShort( 0x2da ), /*
Offset= 730 (904) */
/* 176 */ NdrFcLong( 0x4008 ), /* 16392
*/
/* 180 */ NdrFcShort( 0x2dc ), /*
Offset= 732 (912) */
/* 182 */ NdrFcLong( 0x400d ), /* 16397
*/
/* 186 */ NdrFcShort( 0x2da ), /*
Offset= 730 (916) */
/* 188 */ NdrFcLong( 0x4009 ), /* 16393
*/
/* 192 */ NdrFcShort( 0x2d8 ), /*
Offset= 728 (920) */
/* 194 */ NdrFcLong( 0x6000 ), /* 24576
*/
/* 198 */ NdrFcShort( 0x2d6 ), /*
Offset= 726 (924) */

```

```

/* 200 */ NdrFcLong( 0x400c ), /* 16396
*/
/* 204 */ NdrFcShort( 0x2d4 ), /*
Offset= 724 (928) */
/* 206 */ NdrFcLong( 0x10 ), /* 16 */
/* 210 */ NdrFcShort( 0x8002 ), /*
Simple arm type: FC_CHAR */
/* 212 */ NdrFcLong( 0x12 ), /* 18 */
/* 216 */ NdrFcShort( 0x8006 ), /*
Simple arm type: FC_SHORT */
/* 218 */ NdrFcLong( 0x13 ), /* 19 */
/* 222 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 224 */ NdrFcLong( 0x15 ), /* 21 */
/* 228 */ NdrFcShort( 0x800b ), /*
Simple arm type: FC_HYPER */
/* 230 */ NdrFcLong( 0x16 ), /* 22 */
/* 234 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 236 */ NdrFcLong( 0x17 ), /* 23 */
/* 240 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 242 */ NdrFcLong( 0xe ), /* 14 */
/* 246 */ NdrFcShort( 0x2b2 ), /*
Offset= 690 (936) */
/* 248 */ NdrFcLong( 0x400e ), /* 16398
*/
/* 252 */ NdrFcShort( 0x2b6 ), /*
Offset= 694 (946) */
/* 254 */ NdrFcLong( 0x4010 ), /* 16400
*/
/* 258 */ NdrFcShort( 0x2b4 ), /*
Offset= 692 (950) */
/* 260 */ NdrFcLong( 0x4012 ), /* 16402
*/
/* 264 */ NdrFcShort( 0x270 ), /*
Offset= 624 (888) */
/* 266 */ NdrFcLong( 0x4013 ), /* 16403
*/
/* 270 */ NdrFcShort( 0x26e ), /*
Offset= 622 (892) */
/* 272 */ NdrFcLong( 0x4015 ), /* 16405
*/
/* 276 */ NdrFcShort( 0x26c ), /*
Offset= 620 (896) */
/* 278 */ NdrFcLong( 0x4016 ), /* 16406
*/
/* 282 */ NdrFcShort( 0x262 ), /*
Offset= 610 (892) */
/* 284 */ NdrFcLong( 0x4017 ), /* 16407
*/
/* 288 */ NdrFcShort( 0x25c ), /*
Offset= 604 (892) */
/* 290 */ NdrFcLong( 0x0 ), /* 0 */
/* 294 */ NdrFcShort( 0x0 ), /*
Offset= 0 (294) */
/* 296 */ NdrFcLong( 0x1 ), /* 1 */
/* 300 */ NdrFcShort( 0x0 ), /*
Offset= 0 (300) */
/* 302 */ NdrFcShort( 0xfffff ), /*
Offset= -1 (301) */
/* 304 */
0x15,
/* FC_STRUCT */
0x7,
/* 7 */
/* 306 */ NdrFcShort( 0x8 ), /* 8 */
/* 308 */ 0xb, /*
FC_HYPER */
0x5b,
/* FC_END */
0x12,
0x0, /* FC_UP */
/* 312 */ NdrFcShort( 0xe ), /*
Offset= 14 (326) */

```

```

/* 314 */
0x1b,
/* FC_CARRAY */
0x1,
/* 1 */
/* 316 */ NdrFcShort( 0x2 ), /* 2 */
/* 318 */ 0x9, /* Corr
desc: FC_ULONG */
0x0,
/* 320 */ NdrFcShort( 0xfffc ), /* -4 */
/* 322 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 324 */ 0x6, /*
FC_SHORT */
0x5b,
/* FC_END */
0x17,
/* FC_CSTRUCT */
0x3,
/* 3 */
/* 328 */ NdrFcShort( 0x8 ), /* 8 */
/* 330 */ NdrFcShort( 0xfffff0 ), /*
Offset= -16 (314) */
/* 332 */ 0x8, /*
FC_LONG */
0x8,
/* FC_LONG */
/* 334 */ 0x5c, /*
FC_PAD */
0x5b,
/* FC_END */
0x2f,
/* FC_IP */
0x5a,
/* FC_CONSTANT_IID
*/
/* 338 */ NdrFcLong( 0x0 ), /* 0 */
/* 342 */ NdrFcShort( 0x0 ), /* 0 */
/* 344 */ NdrFcShort( 0x0 ), /* 0 */
/* 346 */ 0xc0, /* 192
*/
0x0,
/* 0 */
/* 348 */ 0x0, /* 0 */
/* 0 */
/* 350 */ 0x0, /* 0 */
/* 0 */
/* 352 */ 0x0, /* 0 */
/* 0x46,
*/
/* 70 */
/* 354 */
0x2f,
/* FC_IP */
0x5a,
/* FC_CONSTANT_IID
*/
/* 356 */ NdrFcLong( 0x20400 ), /*
132096 */
/* 360 */ NdrFcShort( 0x0 ), /* 0 */
/* 362 */ NdrFcShort( 0x0 ), /* 0 */
/* 364 */ 0xc0, /* 192
*/
0x0,
/* 0 */
/* 366 */ 0x0, /* 0 */
/* 0 */
/* 368 */ 0x0, /* 0 */

```

```

0x0,
/* 370 */ 0x0, /* 0 */
/* 372 */
0x10, /* FC_UP [pointer_deref] */
/* 374 */ NdrFcShort( 0x2 ), /*
Offset= 2 (376) */
/* 376 */
0x0, /* FC_UP */
/* 378 */ NdrFcShort( 0x1e4 ), /*
Offset= 484 (862) */
/* 380 */
FC_ENCAPSULATED_UNION /*
/* 137 */
/* 382 */ NdrFcShort( 0x20 ), /* 32 */
/* 384 */ NdrFcShort( 0xa ), /* 10 */
/* 386 */ NdrFcLong( 0x8 ), /* 8 */
/* 390 */ NdrFcShort( 0x50 ), /*
Offset= 80 (470) */
/* 392 */ NdrFcLong( 0xd ), /* 13 */
/* 396 */ NdrFcShort( 0x70 ), /*
Offset= 112 (508) */
/* 398 */ NdrFcLong( 0x9 ), /* 9 */
/* 402 */ NdrFcShort( 0x90 ), /*
Offset= 144 (546) */
/* 404 */ NdrFcLong( 0xc ), /* 12 */
/* 408 */ NdrFcShort( 0xb0 ), /*
Offset= 176 (584) */
/* 410 */ NdrFcLong( 0x24 ), /* 36 */
/* 414 */ NdrFcShort( 0x102 ), /*
Offset= 258 (672) */
/* 416 */ NdrFcLong( 0x800d ), /* 32781
*/
/* 420 */ NdrFcShort( 0x11e ), /*
Offset= 286 (706) */
/* 422 */ NdrFcLong( 0x10 ), /* 16 */
/* 426 */ NdrFcShort( 0x138 ), /*
Offset= 312 (738) */
/* 428 */ NdrFcLong( 0x2 ), /* 2 */
/* 432 */ NdrFcShort( 0x14e ), /*
Offset= 334 (766) */
/* 434 */ NdrFcLong( 0x3 ), /* 3 */
/* 438 */ NdrFcShort( 0x164 ), /*
Offset= 356 (794) */
/* 440 */ NdrFcLong( 0x14 ), /* 20 */
/* 444 */ NdrFcShort( 0x17a ), /*
Offset= 378 (822) */
/* 446 */ NdrFcShort( 0xfffff ), /*
Offset= -1 (445) */
/* 448 */
0x21, /* FC_BOGUS_ARRAY
*/
/* 3 */
0x3,
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
0x0,
/* 454 */ NdrFcShort( 0x0 ), /* 0 */
/* 456 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 458 */ NdrFcLong( 0xfffff ), /* -1 */
/* 462 */ NdrFcShort( 0x0 ), /* Corr
flags: */
/* 464 */

```

```

0x666 /* NdrFcShort( 0xffff74 ), /*
Offset= -140 (326) */
/* 468 */ 0x5c, /*
FC_PAD */
0x12,
/* 470 */
0x5b, /* FC_END */
0x1a,
FC_BOGUS_STRUCT /*
/* 3 */
0x3,
/* 472 */ NdrFcShort( 0x10 ), /* 16 */
/* 474 */ NdrFcShort( 0x0 ), /* 0 */
/* 476 */ NdrFcShort( 0x6 ), /*
Offset= 6 (482) */
/* 478 */ 0x8, /*
FC_LONG */
0x40, /* FC_STRUCTPAD4
*/
/* 480 */ 0x36, /*
FC_POINTER */
0x5b, /* FC_END */
/* 482 */
0x0, /* FC_RP */
/* 484 */ NdrFcShort( 0xfffffdc ), /*
Offset= -36 (448) */
/* 486 */
0x21, /* FC_BOGUS_ARRAY
*/
0x3,
/* 3 */
0x3,
/* 488 */ NdrFcShort( 0x0 ), /* 0 */
/* 490 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
0x0,
/* 492 */ NdrFcShort( 0x0 ), /* 0 */
/* 494 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 496 */ NdrFcLong( 0xfffff ), /* -1 */
/* 500 */ NdrFcShort( 0x0 ), /* Corr
flags: */
/* 502 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
0x0,
/* 504 */ NdrFcShort( 0xfffff58 ), /*
Offset= -168 (336) */
/* 506 */ 0x5c, /*
FC_PAD */
0x5b, /* FC_END */
/* 508 */
0x1a,
FC_BOGUS_STRUCT /*
/* 3 */
0x3,
/* 510 */ NdrFcShort( 0x10 ), /* 16 */
/* 512 */ NdrFcShort( 0x0 ), /* 0 */
/* 514 */ NdrFcShort( 0x6 ), /*
Offset= 6 (520) */
/* 516 */ 0x8, /*
FC_LONG */
0x40, /* FC_STRUCTPAD4
*/
/* 518 */ 0x36, /*
FC_POINTER */

```

```

0x5b, /* FC_END */
0x11,
0x0, /* FC_RP */
/* 522 */ NdrFcShort( 0xfffffdc ), /*
Offset= -36 (486) */
/* 524 */
0x21, /* FC_BOGUS_ARRAY
*/
0x3,
/* 3 */
0x3,
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
0x0,
/* 530 */ NdrFcShort( 0x0 ), /* 0 */
/* 532 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 534 */ NdrFcLong( 0xfffff ), /* -1 */
/* 538 */ NdrFcShort( 0x0 ), /* Corr
flags: */
/* 540 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
0x0,
/* 542 */ NdrFcShort( 0xfffff44 ), /*
Offset= -188 (354) */
/* 544 */ 0x5c, /*
FC_PAD */
0x5b, /* FC_END */
/* 546 */
0x1a,
FC_BOGUS_STRUCT /*
/* 3 */
0x3,
/* 548 */ NdrFcShort( 0x10 ), /* 16 */
/* 550 */ NdrFcShort( 0x0 ), /* 0 */
/* 552 */ NdrFcShort( 0x6 ), /*
Offset= 6 (558) */
/* 554 */ 0x8, /*
FC_LONG */
0x40, /* FC_STRUCTPAD4
*/
/* 556 */ 0x36, /*
FC_POINTER */
0x5b, /* FC_END */
/* 558 */
0x11,
0x0, /* FC_RP */
/* 560 */ NdrFcShort( 0xfffffdc ), /*
Offset= -36 (524) */
/* 562 */
0x21, /* FC_BOGUS_ARRAY
*/
0x3,
/* 3 */
0x3,
/* 564 */ NdrFcShort( 0x0 ), /* 0 */
/* 566 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
0x0,
/* 568 */ NdrFcShort( 0x0 ), /* 0 */
/* 570 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 572 */ NdrFcLong( 0xfffff ), /* -1 */

```

```

/* 576 */ NdrFcShort( 0x0 ), /* Corr
flags: */
/* 578 */
0x0, /* FC_UP */
/* 580 */ NdrFcShort( 0x176 ), /*
Offset= 374 (954) */
/* 582 */ 0x5c, /*
FC_PAD */
/* 584 */ /* FC_END */
0x1a,
FC_BOGUS_STRUCT */
0x3,
/* 586 */ NdrFcShort( 0x10 ), /* 3 */
/* 588 */ NdrFcShort( 0x0 ), /* 16 */
/* 590 */ NdrFcShort( 0x6 ), /* 0 */
Offset= 6 (596) */
/* 592 */ 0x8, /*
FC_LONG */
0x40, /* FC_STRUCTPAD4
*/
/* 594 */ 0x36, /*
FC_POINTER */
0x5b, /* FC_END */
/* 596 */
0x11,
0x0, /* FC_RP */
/* 598 */ NdrFcShort( 0xffffdc ), /*
Offset= -36 (562) */
/* 600 */
0x2f, /* FC_IP */
0x5a, /* FC_CONSTANT_IID
*/
/* 602 */ NdrFcLong( 0x2f ), /* 47 */
/* 606 */ NdrFcShort( 0x0 ), /* 0 */
/* 608 */ NdrFcShort( 0x0 ), /* 0 */
/* 610 */ 0xc0, /* 192
*/
/* 612 */ 0x0, /* 0 */
/* 614 */ 0x0, /* 0 */
/* 616 */ 0x0, /* 0 */
/* 618 */ /* 70 */
0x1b, /* FC_CARRAY */
/* 620 */ NdrFcShort( 0x1 ), /* 0 */
/* 622 */ 0x19, /* 1 */
desc: field pointer, FC_ULONG */
0x0, /*
*/
/* 624 */ NdrFcShort( 0x4 ), /* 4 */
/* 626 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 628 */ 0x1, /*
FC_BYTE */
0x5b, /* FC_END */
/* 630 */

```

```

0x1a,
/*
FC_BOGUS_STRUCT */
0x3,
/* 632 */ NdrFcShort( 0x18 ), /* 3 */
/* 634 */ NdrFcShort( 0x0 ), /* 24 */
/* 636 */ NdrFcShort( 0xa ), /* 0 */
Offset= 10 (646) */
/* 638 */ 0x8, /*
FC_LONG */
0x8, /* FC_LONG */
/* 640 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
0x0,
/* 642 */ NdrFcShort( 0xffffd6 ), /* 0 */
Offset= -42 (600) */
/* 644 */ 0x36, /*
FC_POINTER */
0x5b, /* FC_END */
/* 646 */
0x12,
0x0, /* FC_UP */
/* 648 */ NdrFcShort( 0xfffffe2 ), /*
Offset= -30 (618) */
/* 650 */
0x21, /* FC_BOGUS_ARRAY
*/
0x3,
/* 652 */ NdrFcShort( 0x0 ), /* 3 */
/* 654 */ 0x19, /* 0 */
desc: field pointer, FC_ULONG */
0x0, /*
*/
/* 656 */ NdrFcShort( 0x0 ), /* 0 */
/* 658 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 660 */ NdrFcLong( 0xfffffff ), /* -1 */
/* 664 */ NdrFcShort( 0x0 ), /* Corr
flags: */
/* 666 */
0x12,
0x0, /* FC_UP */
/* 668 */ NdrFcShort( 0xfffffda ), /*
Offset= -38 (630) */
/* 670 */ 0x5c, /*
FC_PAD */
0x5b, /* FC_END */
/* 672 */
0x1a,
FC_BOGUS_STRUCT */
0x3,
/* 674 */ NdrFcShort( 0x10 ), /* 3 */
/* 676 */ NdrFcShort( 0x0 ), /* 16 */
/* 678 */ NdrFcShort( 0x6 ), /* 0 */
Offset= 6 (684) */
/* 680 */ 0x8, /*
FC_LONG */
0x40, /* FC_STRUCTPAD4
*/
/* 682 */ 0x36, /*
FC_POINTER */
0x5b, /* FC_END */
/* 684 */
0x11,
0x0, /* FC_RP */

```

```

/* 686 */ NdrFcShort( 0xfffffdc ), /*
Offset= 36 (650) */
0x1d, /* FC_SMFARRAY */
0x0,
/* 690 */ NdrFcShort( 0x8 ), /* 0 */
/* 692 */ 0x1, /* 8 */
FC_BYTE */
0x5b, /* FC_END */
/* 694 */
0x15, /* FC_STRUCT */
0x3,
/* 696 */ NdrFcShort( 0x10 ), /* 3 */
/* 698 */ 0x8, /* 16 */
FC_LONG */
0x6, /* FC_SHORT */
/* 700 */ 0x6, /*
FC_SHORT */
0x4c,
FC_EMBEDDED_COMPLEX */
/* 702 */ 0x0, /* 0 */
NdrFcShort( 0xfffff1 ), /*
Offset= -15 (688) */
0x5b, /* FC_END */
/* 706 */
0x1a,
FC_BOGUS_STRUCT */
0x3,
/* 708 */ NdrFcShort( 0x20 ), /* 32 */
/* 710 */ NdrFcShort( 0x0 ), /* 0 */
/* 712 */ NdrFcShort( 0xa ), /*
Offset= 10 (722) */
/* 714 */ 0x8, /*
FC_LONG */
0x40, /* FC_STRUCTPAD4
*/
/* 716 */ 0x36, /*
FC_POINTER */
0x4c,
FC_EMBEDDED_COMPLEX */
/* 718 */ 0x0, /* 0 */
NdrFcShort( 0xfffffe7 ), /*
Offset= -25 (694) */
0x5b, /* FC_END */
/* 722 */
0x11,
0x0, /* FC_RP */
/* 724 */ NdrFcShort( 0xfffff12 ), /*
Offset= -238 (486) */
/* 726 */
0x1b, /* FC_CARRAY */
0x0,
/* 728 */ NdrFcShort( 0x1 ), /* 0 */
/* 730 */ 0x19, /* 1 */
desc: field pointer, FC_ULONG */
0x0, /*
*/

```

```

/* 732 */ NdrFcShort( 0x0 ), /* 0 */
/* 734 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 736 */ 0x1, /*
FC_BYTE */

/* 738 */
/* FC_END */
0x5b,
0x1a,
FC_BOGUS_STRUCT */
/*
0x3,
/* 740 */ NdrFcShort( 0x10 ), /* 3 */
/* 742 */ NdrFcShort( 0x0 ), /* 16 */
/* 744 */ NdrFcShort( 0x6 ), /* 0 */
Offset= 6 (750) */
/* 746 */ 0x8, /*
FC_LONG */
/* FC_STRUCTPAD4
0x40,
*/
/* 748 */ 0x36, /*
FC_POINTER */
/* FC_END */
0x5b,
/* 750 */
0x12,
/* FC_UP */
/* 752 */ NdrFcShort( 0xfffff6 ), /*
Offset=-26 (726) */
/* 754 */
0x1b, /* FC_CARRY */
0x1,
/* 1 */
/* 756 */ NdrFcShort( 0x2 ), /* 2 */
/* 758 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
0x0, /*
/* 760 */ NdrFcShort( 0x0 ), /* 0 */
/* 762 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 764 */ 0x6, /*
FC_SHORT */
/* FC_END */
0x5b,
/* 766 */
0x1a,
FC_BOGUS_STRUCT */
/*
0x3,
/* 3 */
/* 768 */ NdrFcShort( 0x10 ), /* 16 */
/* 770 */ NdrFcShort( 0x0 ), /* 0 */
/* 772 */ NdrFcShort( 0x6 ), /*
Offset= 6 (778) */
/* 774 */ 0x8, /*
FC_LONG */
0x40, /* FC_STRUCTPAD4
*/
/* 776 */ 0x36, /*
FC_POINTER */
/* FC_END */
0x5b,
/* 778 */
0x12,
0x0, /* FC_UP */
/* 780 */ NdrFcShort( 0xfffff6 ), /*
Offset=-26 (754) */
/* 782 */
0x1b, /* FC_CARRY */

```

```

/* 784 */ NdrFcShort( 0x43 ), /* 0x3,
/* 786 */ 0x19, /* 4 */
desc: field pointer, FC_ULONG */
0x0,
/* 788 */ NdrFcShort( 0x0 ), /* 0 */
/* 790 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 792 */ 0x8, /*
FC_LONG */
/* FC_END */
0x5b,
/* 794 */
0x1a,
FC_BOGUS_STRUCT */
/*
0x3,
/* 3 */
/* 796 */ NdrFcShort( 0x10 ), /* 16 */
/* 798 */ NdrFcShort( 0x0 ), /* 0 */
/* 800 */ NdrFcShort( 0x6 ), /*
Offset= 6 (806) */
/* 802 */ 0x8, /*
FC_LONG */
/* FC_STRUCTPAD4
0x40,
*/
/* 804 */ 0x36, /*
FC_POINTER */
/* FC_END */
0x5b,
/* 806 */
0x12,
/* FC_UP */
/* 808 */ NdrFcShort( 0xfffff6 ), /*
Offset=-26 (782) */
/* 810 */
0x1b, /* FC_CARRY */
0x7,
/* 7 */
/* 812 */ NdrFcShort( 0x8 ), /* 8 */
/* 814 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
0x0,
/* 816 */ NdrFcShort( 0x0 ), /* 0 */
/* 818 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 820 */ 0xb, /*
FC_HYPER */
/* FC_END */
0x5b,
/* 822 */
0x1a,
FC_BOGUS_STRUCT */
/*
0x3,
/* 3 */
/* 824 */ NdrFcShort( 0x10 ), /* 16 */
/* 826 */ NdrFcShort( 0x0 ), /* 0 */
/* 828 */ NdrFcShort( 0x6 ), /*
Offset= 6 (834) */
/* 830 */ 0x8, /*
FC_LONG */
0x40, /* FC_STRUCTPAD4
*/
/* 832 */ 0x36, /*
FC_POINTER */
/* FC_END */
0x5b,
/* 834 */

```

```

/* 836 */ NdrFcShort( 0xfffff6 ), /* 0x12,
Offset=-26 (810) */
/* 838 */
0x15, /* FC_STRUCT */
0x3,
/* 3 */
/* 840 */ NdrFcShort( 0x8 ), /* 8 */
/* 842 */ 0x8, /*
FC_LONG */
0x8, /* FC_LONG */
/* 844 */ 0x5c, /*
FC_PAD */
/* FC_END */
0x5b,
/* 846 */
0x1b, /* FC_CARRY */
0x3,
/* 3 */
/* 848 */ NdrFcShort( 0x8 ), /* 8 */
/* 850 */ 0x7, /* Corr
desc: FC_USHORT */
0x0,
/* 852 */ NdrFcShort( 0xffc8 ), /* -56 */
/* 854 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 856 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
0x0,
/* 858 */ NdrFcShort( 0xfffffec ), /*
Offset=-20 (838) */
/* 860 */ 0x5c, /*
FC_PAD */
0x5b, /* FC_END */
/* 862 */
0x1a,
FC_BOGUS_STRUCT */
/*
0x3,
/* 3 */
/* 864 */ NdrFcShort( 0x38 ), /* 56 */
/* 866 */ NdrFcShort( 0xfffffec ), /*
Offset=-20 (846) */
/* 868 */ NdrFcShort( 0x0 ), /*
Offset= 0 (868) */
/* 870 */ 0x6, /*
FC_SHORT */
0x6, /* FC_SHORT */
/* 872 */ 0x8, /*
FC_LONG */
0x8, /* FC_LONG */
/* 874 */ 0x40, /*
FC_STRUCTPAD4 */
0x4c,
/*
FC_EMBEDDED_COMPLEX */
/* 876 */ 0x0, /* 0 */
NdrFcShort( 0xfffff0f ), /*
Offset=-497 (380) */
0x5b, /* FC_END */
/* 880 */
0x12,
0x0, /* FC_UP */

```

```

/* 882 */ NdrFcShort( 0xfffff04 ), /*
Offset=-252 (630) */
/* 884 */
0x8, /* FC_UP [simple_pointer] */
/* 886 */ 0x1, /*
FC_BYTE */
0x5c,
/* FC_PAD */
/* 888 */
0x12,
0x8, /* FC_UP [simple_pointer] */
/* 890 */ 0x6, /*
FC_SHORT */
0x5c,
/* FC_PAD */
/* 892 */
0x12,
0x8, /* FC_UP [simple_pointer] */
/* 894 */ 0x8, /*
FC_LONG */
0x5c,
/* FC_PAD */
/* 896 */
0x12,
0x8, /* FC_UP [simple_pointer] */
/* 898 */ 0xb, /*
FC_HYPER */
0x5c,
/* FC_PAD */
/* 900 */
0x12,
0x8, /* FC_UP [simple_pointer] */
/* 902 */ 0xa, /*
FC_FLOAT */
0x5c,
/* FC_PAD */
/* 904 */
0x12,
0x8, /* FC_UP [simple_pointer] */
/* 906 */ 0xc, /*
FC_DOUBLE */
0x5c,
/* FC_PAD */
/* 908 */
0x12,
0x0, /* FC_UP */
/* 910 */ NdrFcShort( 0xffffda2 ), /*
Offset=-606 (304) */
/* 912 */
0x12,
0x10, /* FC_UP [pointer_deref] */
/* 914 */ NdrFcShort( 0xffffda4 ), /*
Offset=-604 (310) */
/* 916 */
0x12,
0x10, /* FC_UP [pointer_deref] */
/* 918 */ NdrFcShort( 0xffffdba ), /*
Offset=-582 (336) */
/* 920 */
0x12,
0x10, /* FC_UP [pointer_deref] */
/* 922 */ NdrFcShort( 0xffffdc8 ), /*
Offset=-568 (354) */
/* 924 */
0x12,
0x10, /* FC_UP [pointer_deref] */
/* 926 */ NdrFcShort( 0xffffdd6 ), /*
Offset=-554 (372) */
/* 928 */
0x12,
0x10, /* FC_UP [pointer_deref] */
/* 930 */ NdrFcShort( 0x2 ), /*
Offset=2 (932) */
/* 932 */

```

```

0x12,
/* 934 */ NdrFcShort( 0x14 ), /*
Offset=20 (954) */
/* 936 */
0x15,
/* FC_STRUCT */
0x7,
/* 7 */
/* 938 */ NdrFcShort( 0x10 ), /* 16 */
/* 940 */ 0x6, /*
FC_SHORT */
0x1,
/* FC_BYTE */
/* 942 */ 0x1, /*
FC_BYTE */
0x8,
/* FC_LONG */
/* 944 */ 0xb, /*
FC_HYPER */
0x5b,
/* FC_END */
/* 946 */
0x12,
0x0, /* FC_UP */
/* 948 */ NdrFcShort( 0xffffff4 ), /*
Offset=-12 (936) */
/* 950 */
0x12,
0x8, /* FC_UP [simple_pointer] */
/* 952 */ 0x2, /*
FC_CHAR */
0x5c,
/* FC_PAD */
/* 954 */
0x1a,
/*
FC_BOGUS_STRUCT */
0x7,
/* 7 */
/* 956 */ NdrFcShort( 0x20 ), /* 32 */
/* 958 */ NdrFcShort( 0x0 ), /* 0 */
/* 960 */ NdrFcShort( 0x0 ), /*
Offset=0 (960) */
/* 962 */ 0x8, /*
FC_LONG */
0x8,
/* FC_LONG */
/* 964 */ 0x6, /*
FC_SHORT */
0x6,
/* FC_SHORT */
/* 966 */ 0x6, /*
FC_SHORT */
0x6,
/* FC_SHORT */
/* 968 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
0x0,
/* 0 */
/* 970 */ NdrFcShort( 0xfffffc3c ), /*
Offset=-964 (6) */
/* 972 */ 0x5c, /*
FC_PAD */
0x5b,
/* FC_END */
/* 974 */ 0xb4, /*
FC_USER_MARSHAL */
0x83,
/* 131 */
/* 976 */ NdrFcShort( 0x0 ), /* 0 */
/* 978 */ NdrFcShort( 0x18 ), /* 24 */
/* 980 */ NdrFcShort( 0x0 ), /* 0 */
/* 982 */ NdrFcShort( 0xfffffc2c ), /*
Offset=-980 (2) */
/* 984 */

```

```

0x11,
/* 986 */ NdrFcShort( 0x0 ), /*
Offset=6 (992) */
/* 988 */
0x13,
0x0, /* FC_OP */
/* 990 */ NdrFcShort( 0xfffffdc ), /*
Offset=-36 (954) */
/* 992 */ 0xb4, /*
FC_USER_MARSHAL */
0x83,
/* 131 */
/* 994 */ NdrFcShort( 0x0 ), /* 0 */
/* 996 */ NdrFcShort( 0x18 ), /* 24 */
/* 998 */ NdrFcShort( 0x0 ), /* 0 */
/* 1000 */ NdrFcShort( 0xffffff4 ), /*
Offset=-12 (988) */
0x0
}
};

static const
USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ] =
{
{
VARIANT_UserSize
,VARIANT_UserMarshal
,VARIANT_UserUnmarshal
,VARIANT_UserFree
}
};

/* Standard interface:
_MIDL_itf_tppc_com_ps_0000, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xc0,0x00,
0x00,0x00,0x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEE6AA2,0x84B1,0x11d2,{0xBA,0x47,
0x00,0xc0,0x4f,0xbf,0xe0,0x8b}} */

#pragma code_seg(".orpc")
static const unsigned short
ITPCC_FormatStringOffsetTable[] =
{
0,
44,
88,
132,
176,
220
};

static const MIDL_STUBLESS_PROXY_INFO
ITPCC_ProxyInfo =
{
&Object_StubDesc,

```

```

__MIDL_ProcFormatString.Format,
&ITPCC_FormatStringOffsetTable[-3],
0,
0,
0
};

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

CINTERFACE_PROXY_VTABLE(9)
_ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    (void *) (INT_PTR) -1 /* ITPCC::NewOrder */
    ,
    (void *) (INT_PTR) -1 /* ITPCC::Payment */ ,
    (void *) (INT_PTR) -1 /* ITPCC::Delivery */ ,
    (void *) (INT_PTR) -1 /* ITPCC::StockLevel */
    ,
    (void *) (INT_PTR) -1 /* ITPCC::OrderStatus
    */ ,
    (void *) (INT_PTR) -1 /*
ITPCC::CallSetComplete */
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,
    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

static const MIDL_STUB_DESC Object_StubDesc
=
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x50002, /* Ndr library version */
    0,
    0x600015b, /* MIDL Version 6.0.347 */
    0,
    UserMarshalRoutines,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* cs routines */
    0, /* proxy/server info */
    0 /* Reserved5 */
};

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_AMD64)*/

```

# Appendix B Database Load

## Setup.vbs

```

-----
'--- FILE:  SETUP.VBS
'---      Microsoft TPC-C Kit Ver. 4.63
'---      Copyright Microsoft, 2001, 2002, 2005,
2004, 2005
'---      All Rights Reserved
'---
'--- PURPOSE:  This module performs the tasks to
create and populate a TPC-C database
'---
-----
'--- set the kit version variable for later display
'---
Kit_Version = " 4.63"
SQL2K_Kit_Version = " 4.55"
'---
'--- open an windows scripting object
'---
Set WshShell = CreateObject("WScript.Shell")
'---
'--- set up windows scripting argument collection
'---
Set ObjArgs = WScript.ARGUMENTS
'---
'--- grab the platform, ia64, x86, from the environment
variables
'---
Platform =
LCase(Left(WshShell.ExpandEnvironmentStrings("%PROCESSOR_IDENTIFIER%", 4))
Select Case Platform
    Case "ia64"
        Platform = "IA64"
    Case Else
        Platform = "X86"
End Select
'---
'--- grab the processor architecture. This is to
determine if the
'--- user is trying to run in 32-bit emulation on a 64-bit
machine.
'--- if that is the case, then display a message and exit.
'---
Proc_Architecture =
WshShell.ExpandEnvironmentStrings("%PROCESSOR_
ARCHITECTURE%")
If Platform = "IA64" And Proc_Architecture = "x86"
Then
    WScript.Echo
    "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    "
    WScript.Echo "!!

    WScript.Echo "!!! You are attempting to run this
SETUP in the 32-bit (WOW) emulation !!!"
    WScript.Echo "!!! mode on an ia64 system.
Please restart the SETUP in a native    !!!"

```

```

    WScript.Echo "!! 64-bit environment.
    !!!
    !!!

    WScript.Echo
    "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    "
    WScript.Quit
End If
'---
'--- before we go any further, make sure that
'--- we are running Windows Scripting Host 5.6
'--- or higher
'---
If WScript.Version < 5.6 Then
    WScript.Echo
    "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    "
    WScript.Echo "!!
    !!!
    WScript.Echo "!!! You do not have the proper
version of the Windows Scripting Host    !!!"
    WScript.Echo "!!! installed. Please install the
latest Windows Scripting Host from    !!!"
    WScript.Echo "!!! ..\tools\wsh\scripten.exe and
restart setup.    !!!"
    WScript.Echo "!!
    !!!
    WScript.Echo
    "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    "
    WScript.Quit
End If
'---
'--- display banner message
'---
WScript.Echo
"*****
*****"
WScript.Echo "*"
*
WScript.Echo "*" Microsoft TPC-C Benchmark Kit Ver."
& Kit_Version & "    *"
WScript.Echo "*"
*
WScript.Echo "*" Database Setup
*
WScript.Echo "*"
*
WScript.Echo
"*****
*****"
'---
'--- Initialize an array of the TPC-C table names
'---
Dim TableArray(9)
TableArray(0) = "warehouse"
TableArray(1) = "district"
TableArray(2) = "customer"
TableArray(3) = "history"
TableArray(4) = "new_order"
TableArray(5) = "orders"
TableArray(6) = "order_line"
TableArray(7) = "item"
TableArray(8) = "stock"
TableArray(9) = "tpccldr"
'---
'--- Initialize an array of the TPC-C build log file names
'---
Dim LogFileArray(21)
LogFileArray(0) = "version.log"
LogFileArray(1) = "removedb.log"
LogFileArray(2) = "createdb.log"
LogFileArray(3) = "tables.log"

```

```

LogFileArray(4) = "tblsect.log"
LogFileArray(5) = "tbltbl.log"
LogFileArray(6) = "idxtblcl.log"
LogFileArray(7) = "idxwarcl.log"
LogFileArray(8) = "idxcuscl.log"

LogFileArray(9) = "idxnodcl.log"
LogFileArray(10) = "idxdiscl.log"
LogFileArray(11) = "idxstklcl.log"
LogFileArray(12) = "idxodlcl.log"
LogFileArray(13) = "idxcusnc.log"
LogFileArray(14) = "idxhisccl.log"
LogFileArray(15) = "idxordnc.log"
LogFileArray(16) = "bulkload.log"
LogFileArray(17) = "dbopt2.log"
LogFileArray(18) = "nurand_load.log"
LogFileArray(19) = "backupdev.log"
LogFileArray(20) = "backupdev.log"
LogFileArray(21) = "verifyload.log"
'---
'--- open a file system object
'---
Set fs = CreateObject("Scripting.FileSystemObject")
'---
'---
'--- first see if the user passed a "?" as the first
parameter
'--- if they did, then show the usage data
'---
If ObjArgs.Length > 0 Then
    If ObjArgs(0) = "?" or ObjArgs(0) = "/"?
    Then
        Call ShowUsage
    End If
'---
'--- get the user passed in parameters
If WScript.Arguments.Named.Exists("S") Then
    ServerName = WScript.Arguments.Named.Item("S")
    If ServerName = "" Then
        ServerName =
WshShell.ExpandEnvironmentStrings("%COMPUTERNA
ME%")
    End If
    figServerName = 1
Else
    ServerName =
WshShell.ExpandEnvironmentStrings("%COMPUTERNA
ME%")
    figServerName = 0
End If
SQLUserID = WScript.Arguments.Named.Item("U")
If SQLUserID <> "" Then
    figSQLUserID = 1
Else
    figSQLUserID = 0
End If
SQLPassword = WScript.Arguments.Named.Item("P")
If SQLPassword <> "" Then
    figSQLPassword = 1
Else
    figSQLPassword = 0
End If
If SQLPassword = "BLANK" Then
    figSQLPassword = 1
    SQLPassword = ""
End If
NumberWarehouses =
WScript.Arguments.Named.Item("W")
If NumberWarehouses <> "" Then
    figNumberWarehouses = 1
Else
    figNumberWarehouses = 0
End If
BuildOption = WScript.Arguments.Named.Item("B")

```



```

If BuildOption <> "" Then
    'validate the build option the user passed in
    BuildOption = LCase(BuildOption)
    Select Case BuildOption
        Case
            "full","bulddb","objects","objectsfull","bulkload","bulkloadfull","backup"

            flgBuildOption = 1
        Case Else
            flgBuildOption = 0
        End Select
    Else
        flgBuildOption = 0
    End If
    DatabaseType = wScript.Arguments.Named.Item("D")
    If DatabaseType <> "" Then
        DatabaseType = LCase(DatabaseType)
        Select Case DatabaseType
            Case "normal","scale_down"
                If DatabaseType = "normal" Then
                    DatabaseType = 0
                Else
                    DatabaseType = 1
                End If
                flgDatabaseType = 1
            Case Else
                flgDatabaseType = 0
        End Select
    Else
        flgDatabaseType = 0
    End If
    UnattendedBuild =
    wScript.Arguments.Named.Item("V")
    If UnattendedBuild <> "" Then
        UnattendedBuild = LCase(UnattendedBuild)
        Select Case UnattendedBuild
            Case "true","false"
                flgUnattendedBuild = 1
            Case Else
                flgUnattendedBuild = 0
        End Select
    Else
        flgUnattendedBuild = 0
    End If
    '--- if something is missing, prompt the user for it
    If flgServerName = 0 Then
        ServerName = GetUserInput("ServerName")
        If ServerName = "" Then
            ServerName =
            WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%")
        End If
    End If
    If flgSQLUserID = 0 Then
        SQLUserID = GetUserInput("SQLUserID")
    End If
    If flgSQLPassword = 0 Then
        SQLPassword = GetUserInput("SQLPassword")
    End If
    If flgNumberWarehouses = 0 Then
        NumberWarehouses =
        GetUserInput("NumberWarehouses")
    End If
    If flgBuildOption = 0 Then
        BuildOption = GetUserInput("BuildOption")
    End If
    If flgDatabaseType = 0 Then
        DatabaseType = (GetUserInput("DatabaseType"))
    End If
    If flgUnattendedBuild = 0 Then
        UnattendedBuild =
        GetUserInput("UnattendedBuild")
    End If

```

```

If SQLPassword = "" Then
    End If
'-----
'--- if the user specified a scale down database, then
show
'--- them the warning message
'-----
If DatabaseType = 1 Then
    MsgBox "WARNING!" & Chr(13) & "The
Scale_Down option is to be used for functional testing
only." _
        & Chr(13) & "The use of this option
will not produce a valid TPC-C result." _
        vbExclamation, "Scale-Down
Warning"
    End If
'-----
'--- before we start to do anything, verify the input
'-----
Select Case BuildOption
    Case "full"
        strBuildOpt = "Full build"
    Case "bulddb"
        strBuildOpt = "Build database only"
    Case "objects"
        strBuildOpt = "Install stored procedures
only"
    Case "objectsfull"
        strBuildOpt = "Install stored procedures
and complete build process"
    Case "bulkload"
        strBuildOpt = "Load data only"
    Case "bulkloadfull"
        strBuildOpt = "Load data and complete
build process"
    Case "backup"
        strBuildOpt = "Backup database"
End Select
If DatabaseType = 1 Then
    strDBType = "Scale Down"
Else
    strDBType = "Normal"
End If
If UnattendedBuild = "false" Then
    rc = MsgBox("The following options will be
used:" & Chr(13) & Chr(10) & Chr(13) & Chr(10) _
        & "Database
Server Name: " & ServerName & Chr(13) & Chr(10)
_
        & Chr(13) &
Chr(10) _
        & "SQL Server
User ID: " & SQLUserID & Chr(13) & Chr(10) _
        & Chr(13) &
Chr(10) _
        & "SQL Server
Password: " & SQLPassword & Chr(13) & Chr(10)
_
        & Chr(13) &
Chr(10) _
        & "Number of
Warehouses: " & NumberWarehouses & Chr(13) &
Chr(10) _
        & Chr(13) &
Chr(10) _
        & "Build
Option: " & strBuildOpt & Chr(13) &
Chr(10) _
        & Chr(13) &
Chr(10) _
        & "Build Type:
" & strDBType & Chr(13) & Chr(10) _

```

```

& "
", 65, If rc = 2 Then
    wScript.Echo ""
    wScript.Echo "TPC-C Setup cancelled by
user."
    wScript.Quit
End If
End If
'-----
'--- parse the ServerName to determine if this is a
named instance
'-----
Slash_Loc = InStr(1,ServerName,"\")
If Slash_Loc <> 0 Then
    SQLInstanceName =
    Right(ServerName,(LEN(ServerName) - Slash_Loc))
Else
    SQLInstanceName = ""
End If
'-----
'--- now we need to figure out if this is SQL Server
2000 or SQL Server 2005
'--- if this is being installed on SQL Server 2000, then
abort the load and
'--- direct the user to use the 4.55 kit which is SQL
Server 2000 compliant
'-----
'If SQLInstanceName = "" Then
    '--- check for default installations
    '--- SQL Server 2000 Default Instance
    strRegKey =
    "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQL
Server\CurrentVersion\CurrentVersion"
    ' If CheckRegKey(strRegKey) = True Then
    ' SQLServerVersionRegKey =
    WshShell.RegRead(strRegKey)
    ' End If
    '--- SQL Server 2005 Default Instance
    strRegKey =
    "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQL
Server\MSSQLServer\CurrentVersion\CurrentVersion"
    ' If CheckRegKey(strRegKey) = True Then
    ' SQLServerVersionRegKey =
    WshShell.RegRead(strRegKey)
    ' End If
    'Else
    '--- SQL Server 2000 Named Instance
    'If
    CheckRegKey("HKEY_LOCAL_MACHINE\SOFTWARE\Mi
crosoft\MSSQLServer\CurrentVersion\CurrentVersion")
= True Then
    ' SQLServerVersionRegKey =
    WshShell.RegRead("HKEY_LOCAL_MACHINE\SOFTWA
RE\Microsoft\MSSQLServer\CurrentVersion\CurrentVer
sion")
    'End If
    '--- SQL Server 2005 Named Instance
    strRegKey =
    "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Micro
soft SQL Server\" & SQLInstanceName &
"\MSSQLServer\CurrentVersion\CurrentVersion"
    ' If CheckRegKey(strRegKey) = True Then
    ' SQLServerVersionRegKey =
    WshShell.RegRead(strRegKey)
    ' End If
    'End If
    'If Left(SQLServerVersionRegKey,1) = "8" Then
    ' SQLServerVersion = "2000"

```



```

Call WriteBuildLog("TPC-C database
created", "")
End If
'----- build tables and stored procedures
'-----
If (BuildOption = "full" Or BuildOption = "objects" Or
BuildOption = "objectsfull") Then
    wScript.Echo ""
    wScript.Echo FormatDateTime(Now,0) & " ==>
Creating TPC-C database tables..."
    Call WriteBuildLog("Create dynamic TPC-
C database tables", "")
    Set oExec = WshShell.Exec("osql -U" &
SQLUserID & " -P" & SQLPassword & " -S" &
ServerName & " -e -i" & DDLDirectory & "Tables.sql -
o" & LogDirectory & "Tables.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"Tables.log")
    If rc <> 0 Then
        Call WriteBuildLog("Create
TPC-C database tables", "Step failed! - Check
Tables.log")
        wScript.Echo
        FormatDateTime(Now,0) & " ==> Creation of TPC-C
database tables failed!! Check Tables.log."
        wScript.Quit
    End If
    wScript.Echo FormatDateTime(Now,0) &
" ==> TPC-C database tables created."
    Call WriteBuildLog("TPC-C database
tables created", "")
    wScript.Echo ""
    wScript.Echo FormatDateTime(Now,0) & " ==>
Installing TPC-C stored procedures..."
    Call WriteBuildLog("Install TPC-C stored
procedures", "")
    wScript.Echo "                New
Order..."
    Call WriteBuildLog("    Install TPC-C
stored procedures (New Order)", "")
    Set oExec = WshShell.Exec("osql -U" &
SQLUserID & " -P" & SQLPassword & " -S" &
ServerName & " -e -i" & DMLDirectory & "neword.sql -
o" & LogDirectory & "SP_NewOrd.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"SP_NewOrd.log")
    If rc <> 0 Then
        Call WriteBuildLog("Install
TPC-C stored procedures (New Order)", "Step failed! -
Check SP_NewOrd.log")
        wScript.Quit
    End If
    wScript.Echo "                New
Order (New)..."
    Call WriteBuildLog("    Install TPC-C
stored procedures (New Order (New))", "")
    Set oExec = WshShell.Exec("osql -U" &
SQLUserID & " -P" & SQLPassword & " -S" &
ServerName & " -e -i" & DMLDirectory &
"TPCC_NEWORDER_NEW.SQL -o" & LogDirectory &
"SP_NewOrd_New.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"SP_NewOrd_New.log")

```

```

    If rc <> 0 Then
        Call WriteBuildLog("Install
TPC-C stored procedures (New Order (New))", "Step
failed! - Check SP_NewOrd_New.log")
        wScript.Quit
    End If
    wScript.Echo "
Payment..."
    Call WriteBuildLog("    Install TPC-C
stored procedures (Payment)", "")
    Set oExec = WshShell.Exec("osql -U" &
SQLUserID & " -P" & SQLPassword & " -S" &
ServerName & " -e -i" & DMLDirectory & "payment.sql
-o" & LogDirectory & "SP_Payment.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"SP_Payment.log")
    If rc <> 0 Then
        Call WriteBuildLog("Install
TPC-C stored procedures (Payment)", "Step failed! -
Check SP_Payment.log")
        wScript.Quit
    End If
    wScript.Echo "                Order
Status..."
    Call WriteBuildLog("    Install TPC-C
stored procedures (Order Status)", "")
    Set oExec = WshShell.Exec("osql -U" &
SQLUserID & " -P" & SQLPassword & " -S" &
ServerName & " -e -i" & DMLDirectory & "ordstat.sql -
o" & LogDirectory & "SP_OrdStat.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"SP_OrdStat.log")
    If rc <> 0 Then
        Call WriteBuildLog("Install
TPC-C stored procedures (Order Status)", "Step failed!
- Check SP_OrdStat.log")
        wScript.Quit
    End If
    wScript.Echo "
Delivery..."
    Call WriteBuildLog("    Install TPC-C
stored procedures (Delivery)", "")
    Set oExec = WshShell.Exec("osql -U" &
SQLUserID & " -P" & SQLPassword & " -S" &
ServerName & " -e -i" & DMLDirectory & "delivery.sql -
o" & LogDirectory & "SP_Delivery.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"SP_Delivery.log")
    If rc <> 0 Then
        Call WriteBuildLog("Install
TPC-C stored procedures (Delivery)", "Step failed! -
Check SP_Delivery.log")
        wScript.Quit
    End If
    wScript.Echo "                Stock
Level..."
    Call WriteBuildLog("    Install TPC-C
stored procedures (Stock Level)", "")
    Set oExec = WshShell.Exec("osql -U" &
SQLUserID & " -P" & SQLPassword & " -S" &
ServerName & " -e -i" & DMLDirectory & "stocklev.sql -
o" & LogDirectory & "SP_StockLev.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop

```

```

    rc = CheckSQLOutput(LogDirectory &
"SP_StockLev.log")
    If rc <> 0 Then
        Call WriteBuildLog("Install
TPC-C stored procedures (Stock Level)", "Step failed! -
Check SP_StockLev.log")
        wScript.Quit
    End If
    wScript.Echo "                Version
(Internal)..."
    Call WriteBuildLog("    Install TPC-C
stored procedures (Version)", "")
    Set oExec = WshShell.Exec("osql -U" &
SQLUserID & " -P" & SQLPassword & " -S" &
ServerName & " -e -i" & DMLDirectory & "version.sql -
o" & LogDirectory & "SP_Version.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"SP_Version.log")
    If rc <> 0 Then
        Call WriteBuildLog("Install
TPC-C stored procedures (Version)", "Step failed! -
Check SP_Version.log")
        wScript.Quit
    End If
    wScript.Echo FormatDateTime(Now,0) &
" ==> TPC-C stored procedures installed."
    Call WriteBuildLog("TPC-C stored
procedures installed", "")
    wScript.Echo ""
End If
If (BuildOption = "full" Or BuildOption = "objectsfull"
Or BuildOption = "bulkload" Or BuildOption =
"bulkloadfull") Then
    wScript.Echo FormatDateTime(Now,0) & " ==>
Setting database options before load..."
    Call WriteBuildLog("Set pre-load
database options (DBOPT1)", "")
    Set oExec = WshShell.Exec("osql -U" &
SQLUserID & " -P" & SQLPassword & " -S" &
ServerName & " -e -i" & UtilityDirectory & "dbopt1.sql -
o" & LogDirectory & "Database_Options_1.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"Database_Options_1.log")
    If rc <> 0 Then
        Call WriteBuildLog("Set pre-
load database options (DBOPT1)", "Step failed! - Check
Database_Options_1.log")
        wScript.Quit
    End If
    wScript.Echo FormatDateTime(Now,0) &
" ==> Pre-load database options set."
    Call WriteBuildLog("Pre-load database
options set", "")
    wScript.Echo ""
'----- before we start tpccldr.exe, check the registry
'----- to ensure that the Shared Memory Protocol is
off.
'----- if it is on, store the setting so we can return
'----- the system to the pre-tpccldr state.
'-----
If
CheckRegKey("HKEY_LOCAL_MACHINE\SOFTWARE\Mi
crosoft\MSSQLServer\Client\SharedMemoryOn") =
True Then
    SharedMemoryRegKey =
WshShell.RegRead("HKEY_LOCAL_MACHINE\SOFTWA
RE\Microsoft\MSSQLServer\Client\SharedMemoryOn")

```

```

1 Then
    If SharedMemoryRegKey =
        WshShell.RegWrite
        "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQL
        Server\ClientSharedMemoryOn", 0, "REG_DWORD"
    Else
        SharedMemoryRegKey = 0
    End If
    Call WriteBuildLog("Loading database
    and creating indexes", "")
    wScript.Echo FormatDateTime(Now, 0) &
    " ==> Loading database and creating indexes..."
    wScript.Echo FormatDateTime(Now, 0) &
    " ==> (This runs in a separate, minimized window.)"
    wScript.Echo ""
    CMD_String = ""
    Select Case Platform
        Case "IA64"
            CMD_String = SetupDirectory &
            "loader\bin\ia64\tpccldr.exe"
        Case Else
            CMD_String = SetupDirectory &
            "loader\bin\x86\tpccldr.exe"
    End Select
    CMD_String = CMD_String & " -S" &
    ServerName
    CMD_String = CMD_String & " -U" &
    SQLUserID
    CMD_String = CMD_String & " -P" &
    SQLPassword
    CMD_String = CMD_String & " -W" &
    NumberWarehouses
    CMD_String = CMD_String & " -f" &
    LogDirectory & "bulkload.log"
    CMD_String = CMD_String & " -L" &
    LogDirectory
    CMD_String = CMD_String & " -d" &
    DDLDirectory
    CMD_String = CMD_String & " -c" &
    DatabaseType
    oExec = WshShell.Run(CMD_String, 2, True)
    If oExec <> 0 Then
        wScript.Echo FormatDateTime(Now, 0) & " ==>
        The TPCCCLR.EXE encountered an error."
        wScript.Echo FormatDateTime(Now, 0) & " ==>
        Check the TPCCCLR.ERR log file for details."
        wScript.Quit
    End If
    '-----
    '--- now that the loader is finished, put the
    '--- SharedMemoryOn registry key back to its
    original
    '--- value.
    '-----
    If SharedMemoryRegKey = 1 Then
        WshShell.RegWrite
        "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQL
        Server\ClientSharedMemoryOn", 1, "REG_DWORD"
    End If
    '-----
    '--- set post-load options
    '-----
    wScript.Echo FormatDateTime(Now, 0) &
    " ==> Setting database options after load..."
    Call WriteBuildLog("Set post-load
    database options (DBOPT2)", "")
    Set oExec = WshShell.Exec("osql -U" &
    SQLUserID & " -P" & SQLPassword & " -S" &
    ServerName & " -e -i" & UtilityDirectory & "dbopt2.sql -
    o" & LogDirectory & "Database_Options_2.log")
    Do While oExec.Status = 0

```

```

    Loop
        wScript.Sleep 100
        rc = CheckSQLOutput(LogDirectory &
        "Database_Options_2.log")
        If rc <> 0 Then
            Call WriteBuildLog("Set
            post-load database options
            (Database_Options_2)","Step failed! - Check
            Database_Options_2.log")
            wScript.Quit
        End If
        wScript.Echo FormatDateTime(Now, 0) &
        " ==> Post-load database options set."
        Call WriteBuildLog("Post-load database
        options set", "")
        wScript.Echo ""
        Call WriteBuildLog("Database load and
        index creation complete", "")
        wScript.Echo FormatDateTime(Now, 0) &
        " ==> Database load and index creation complete."
        wScript.Echo ""
        '-----
        '--- now parse the index creation logs
        '--- to see if there were any errors
        '--- there.
        '-----
        For i = 5 To 15
            rc = CheckSQLOutput(LogDirectory &
            LogFileArray(i))
            If rc <> 0 Then
                wScript.Quit
            End If
        Next
        wScript.Echo FormatDateTime(Now, 0) &
        " ==> Calculating initial database space usage...."
        Call WriteBuildLog("Calculate TPC-C
        initial database space usage", "")
        Set oExec = WshShell.Exec("osql -U" &
        SQLUserID & " -P" & SQLPassword & " -S" &
        ServerName & " -e -i" & ACIDDirectory &
        "space/scripts\spused.sql -o" & LogDirectory &
        "spused.ver")
        Do While oExec.Status = 0
            wScript.Sleep 100
        Loop
        Set oExec = WshShell.Exec("osql -U" &
        SQLUserID & " -P" & SQLPassword & " -S" &
        ServerName & " -e -i" & ACIDDirectory &
        "space/scripts\splog.sql -o" & LogDirectory &
        "splog.ver")
        Do While oExec.Status = 0
            wScript.Sleep 100
        Loop
        Set oExec = WshShell.Exec("osql -U" &
        SQLUserID & " -P" & SQLPassword & " -S" &
        ServerName & " -e -i" & ACIDDirectory &
        "space/scripts\spfiles.sql -o" & LogDirectory &
        "spfiles.ver")
        Do While oExec.Status = 0
            wScript.Sleep 100
        Loop
        wScript.Echo FormatDateTime(Now, 0) &
        " ==> Initial database space usage calculated."
        Call WriteBuildLog("TPC-C initial
        database space usage calculated", "")
        wScript.Echo ""
        '-----
        '--- now that the loader is finished
        '--- check the .err files and if they
        '--- are of zero length, delete them.
        '-----
        Set fsErr =
        CreateObject("Scripting.FileSystemObject")
        Set fErr = fsErr.GetFolder(LogDirectory)

```

```

    Set Each f1 In fErr
        If f1.Type = "ERR File" Then
            If f1.Size = 0 Then
                f1.Delete
            End If
        End If
    Next
    Set fcErr = Nothing
    Set fErr = Nothing
    Set fsErr = Nothing
End If
If (BuildOption = "full" Or BuildOption = "objectsfull"
Or BuildOption = "bulkloadfull" Or BuildOption =
"backup") Then
    wScript.Echo FormatDateTime(Now, 0) & " ==>
    Creating backup device(s)..."
    Call WriteBuildLog("Creating backup
    device(s)", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
    & " -P" & SQLPassword & " -S" & ServerName & " -e -
    i" & DBDirectory & "backupdev.sql -o" & LogDirectory
    & "backupdev.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
    "backupdev.log")
    If rc <> 0 Then
        Call WriteBuildLog("Creating backup
        device(s)","Step failed! - Check backupdev.log")
        wScript.Echo FormatDateTime(Now, 0)
        & " ==> Backup device(s) creation failed! Check
        backupdev.log."
        wScript.Quit
    End If
    wScript.Echo FormatDateTime(Now, 0) & " ==>
    Backup device(s) created."
    Call WriteBuildLog("Backup device(s)
    created", "")
    wScript.Echo ""
    wScript.Echo FormatDateTime(Now, 0) & " ==>
    Backing up TPC-C database...."
    Call WriteBuildLog("Backing up TPC-C
    database", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
    & " -P" & SQLPassword & " -S" & ServerName & " -e -
    i" & DBDirectory & "backup.sql -o" & LogDirectory &
    "backup.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory & "backup.log")
    If rc <> 0 Then
        Call WriteBuildLog("Backing up TPC-C
        database","Step failed! - Check backup.log")
        wScript.Echo FormatDateTime(Now, 0)
        & " ==> Database backup failed! Check backup.log."
        wScript.Quit
    End If
    wScript.Echo FormatDateTime(Now, 0) & " ==>
    Database backup complete."
    wScript.Echo ""
    Call WriteBuildLog("Database backup
    complete", "")
End If
If (BuildOption = "full" Or BuildOption = "objectsfull"
Or BuildOption = "bulkloadfull") Then
    '-----
    '--- run a data load verification script
    '-----
    wScript.Echo FormatDateTime(Now, 0) &
    " ==> Verify initial TPC-C database load...."

```

```

Call WriteBuildLog("Verify TPC-C initial
database load", "")
Set oExec = WshShell.Exec("osql -U" &
SQLUserID & " -P" & SQLPassword & " -S" &
ServerName & " -e -w300 -i" & UtilityDirectory &
"VerifyTPCCLoad.sql -o" & LogDirectory &
"VerifyTPCCLoad.log")
Do While oExec.Status = 0
    wScript.Sleep 100
Loop
rc = CheckSQLOutput(LogDirectory &
"VerifyTPCCLoad.log")
If rc <> 0 Then
    Call WriteBuildLog("Verify TPC-C initial database
load", "Step failed! - Check VerifyTPCCLoad.log")
    wScript.Echo FormatDateTime(Now, 0)
    & " ==> TPC-C database verification failed! Check
VerifyTPCCLoad.log."
    wScript.Quit
End If
wScript.Echo FormatDateTime(Now, 0) &
" ==> TPC-C initial database load verified."
Call WriteBuildLog("TPC-C initial
database load verified", "")
wScript.Echo ""

End If
'-----
'--- display banner message
'-----

wScript.Echo
"*****"
wScript.Echo "*"
wScript.Echo "*"
wScript.Echo "Microsoft TPC-C Benchmark Kit Ver."
& Kit_Version & "

wScript.Echo "*"
wScript.Echo "Database Setup Complete"
wScript.Echo "*"
wScript.Echo "*"
wScript.Echo
"*****"
wScript.Quit
'-----
'--- ShowUsage
'-----
Function ShowUsage()
    wScript.Echo
    "*****"
    wScript.Echo "*"
    wScript.Echo "Microsoft TPC-C Benchmark Kit
Ver." & Kit_Version & "
    wScript.Echo "*"
    wScript.Echo "Usage:
    wScript.Echo "Optionally, you can pass the
following positional arguments to SETUP *"
    wScript.Echo " /S:<Server Name> (can be
***** for local host)
    wScript.Echo " /U:<SQL Server User ID>
(recommended you use sa)
    wScript.Echo " /P:<SQL Sever Account
Password>
    wScript.Echo " (enter BLANK if you do
not have a password defined)
    wScript.Echo " /W:<Number of
Warehouses to Build>

```

```

wScript.Echo " /B:<Build Option>
wScript.Echo "
[full, builddb, objects, objectsfull, bulkload, bulkloadfull, ba
ckup]
wScript.Echo " /D:<Database Type>
wScript.Echo " [normal or scale_down]
wScript.Echo " /V:<Verbose>\Unattended
Install>
wScript.Echo " [true or false]
wScript.Echo "
wScript.Echo " If you do not pass the
parameters, then you will be prompted for the
application.
wScript.Echo "
wScript.Echo
"*****"
wScript.Quit
End Function
'-----
'--- define function to check for any error messages
'-----
Function CheckSQLOutput(SQL_Out)
    ErrorFlag = 0
    Set SQL_fso =
CreateObject("Scripting.FileSystemObject")
    If SQL_fso.FileExists(SQL_Out) Then
        Set SQL_Out_File =
SQL_fso.OpenTextFile(SQL_Out, 1)
        Do While SQL_Out_File.AtEndOfStream <>
True
            SQL_Line = SQL_Out_File.ReadLine
            'first check to see if the output
contains a message about the login password
            If InStr(SQL_Line, "Login failed")
Then
                'display the messages and get
out of here
                ErrorFlag = 1
                wScript.Echo "The login for
userid 'sa' failed."
                wScript.Echo "Please restart
SETUP with the correct password."
            Else
                If InStr(SQL_Line, "Msg") Then
                    'find out where the "Msg"
indicator is in the line
                    LocMsg = InStr(SQL_Line,
"Msg")
                    'find out where the comma
is after the error code
                    LocComma =
InStr(SQL_Line, ",")
                    'now isolate the error code
                    ErrorCode =
Mid(SQL_Line, (LocMsg + 4), (LocComma - (LocMsg +
4)))
                    Select Case ErrorCode
                        Case " 170"
                            ErrorFlag = 1
                            wScript.Echo
"Syntax Error."
                            wScript.Echo
"SQL Server Error 170."

```

```

wScript.Echo
wScript.Echo
"Check CREATEDB.SQL."
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
Case "1801"
    ErrorFlag = 1
    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
wScript.Echo
"Database 'tpcc' already exists."
wScript.Echo
"SQL Server Error 1801."
wScript.Echo
"Check CREATEDB.SQL."
wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
Case "1802"
    ErrorFlag = 1
    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
wScript.Echo
"CREATE DATABASE failed."
wScript.Echo
"SQL Server Error 1802."
wScript.Echo
"Check CREATEDB.SQL."
wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
Case "1921"
    ErrorFlag = 1
    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
wScript.Echo
"CREATE INDEX failed."
wScript.Echo
"SQL Server Error 1921."
wScript.Echo
"Check " & SQL_Out & ".
wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
Case "3013"
    ErrorFlag = 1
    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
wScript.Echo
"BACKUP DATABASE is terminating abnormally."
wScript.Echo
"SQL Server Error 3013."
wScript.Echo
"Check the SQL Server error log for more details."
wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
Case "3201"
    ErrorFlag = 1
    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
wScript.Echo
"Cannot open backup device."
wScript.Echo
"Device error or device off-line."
wScript.Echo
"SQL Server Error 3201."
wScript.Echo
"See the SQL Server error log for more details."
wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
Case "5105"
    ErrorFlag = 1
    wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
wScript.Echo
"Device Activation Error."
wScript.Echo
"SQL Server Error 5105."

```



```

"backup"          TempResponse =
                  Flag = 1
                  Case Else
                    rc = MsgBox("Invalid
Database Build Option.", 21)
                    If rc = 2 Then
                      wScript.Echo ""
                      wScript.Echo
"TPC-C Setup cancelled by user."
                      wScript.Quit
                    End If
                    Flag = 0
                    TempResponse =
InputBox("Build Option:" & Chr(13) &
"(full,builddb,objects,objectsfull,bulkload,bulkloadfull,b
ackup)", , "full")
                    TempResponse =
LCase(TempResponse)
                    End Select
                    Loop
                    Case "DatabaseType"
                      TempResponse =
InputBox("Database Type:" & Chr(13) & "(normal or
scale_down)", "TPC-C Setup (V" & Kit_Version & ")",
"normal")
                      TempResponse =
LCase(TempResponse)
                      '--- set flag
                      Flag = 0
                      Do While Flag = 0
                        Select Case TempResponse
                          Case "normal"
                            TempResponse =
"0"
                            Flag = 1
                            Case "scale_down"
                              TempResponse =
"1"
                              Flag = 1
                              Case Else
                                rc = MsgBox("Invalid
Database Type.", 21)
                                If rc = 2 Then
                                  wScript.Echo ""
                                  wScript.Echo
"TPC-C Setup cancelled by user."
                                  wScript.Quit
                                End If
                                Flag = 0
                                TempResponse =
InputBox("Database Type:" & Chr(13) & "(normal or
scale_down)", , "normal")
                                TempResponse =
LCase(TempResponse)
                                End Select
                                Loop
                                Case "UnattendedBuild"
                                  TempResponse =
InputBox("Unattended Build?:" & Chr(13) & "(true or
false)", "TPC-C Setup (V" & Kit_Version & ")", "false")
                                  TempResponse =
LCase(TempResponse)
                                  '--- set flag
                                  Flag = 0
                                  Do While TempResponse = ""
                                    rc = MsgBox("You must enter
true or false for Unattended Build.", 21)
                                    If rc = 2 Then
                                      wScript.Echo ""

```

```

wScript.Echo "TPC-C
wScript.Quit
End If
TempResponse =
InputBox("Unattended Build?:" & Chr(13) & "(true or
false)", "TPC-C Setup (V" & Kit_Version & ")", "false")
TempResponse =
LCase(TempResponse)
Loop
End Select
GetUserInput = TempResponse
End Function
'-----
'--- end function
'-----
'--- subroutine to write BuildLog data
'-----
Sub WriteBuildLog(StepMessage, ErrorMessage)
  Set StepLog =
fs.OpenTextFile(BuildStepLogFile, 8, true)
  If LEN(ErrorMessage) > 0 Then
    msg =
FormatDateTime(Now,0) & " ==> " & StepMessage &
": " & ErrorMessage
    Else
      msg =
FormatDateTime(Now,0) & " ==> " & StepMessage
    End If
    StepLog.WriteLine (msg)
    StepLog.close
  End Sub
'-----
'--- end sub
'-----
'--- function to check for registry key existence
'-----
Function CheckRegKey(RegStr)
  On Error Resume Next
  WshShell.RegRead RegStr
  If Err Then
    = False
    Else
      = True
    End If
  On Error Goto 0
End Function
'-----
'--- end function
'-----
tpcc.h
// File: TPCC.H
// Microsoft TPC-C Kit Ver. 4.51
// Copyright Microsoft, 1996, 1997, 1998,
1999, 2000, 2001, 2002, 2003, 2005
// Purpose: Header file for TPC-C
database loader
// Build number of TPC Benchmark Kit
#define TPCKIT_VER "4.51"

```

```

// 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 <sqlext.h>
#include <odbcss.h>

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

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

// Default loader arguments
#define BATCH 10000
#define DEFLDPAKSIZE 32768
#define LOADER_RES_FILE
"C:\MSTPCC.450\SETUP\LOGS\load.o
ut"
#define LOADER_LOG_PATH
"C:\MSTPCC.450\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 tables
BOOL      table_item;
// set if loading ITEM table specifically
loading  table_warehouse; // set if
WAREHOUSE, DISTRICT, and STOCK
HISTORY  table_customer;
// set if loading CUSTOMER and
loading  table_orders; // set if
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;

} TPCCLDR_ARGS;

// String length constants
#define SERVER_NAME_LEN      20
#define DATABASE_NAME_LEN   20
#define USER_NAME_LEN       20
#define PASSWORD_LEN        20
#define TABLE_NAME_LEN     20
#define I_DATA_LEN          50
#define I_NAME_LEN          24
#define BRAND_LEN           1
#define LAST_NAME_LEN       16
#define W_NAME_LEN          10
#define ADDRESS_LEN         20
#define STATE_LEN           2
#define ZIP_LEN              9
#define S_DIST_LEN          24
#define S_DATA_LEN          50
#define D_NAME_LEN          10
#define FIRST_NAME_LEN      16
#define MIDDLE_NAME_LEN     2
#define PHONE_LEN           16
#define CREDIT_LEN          2
#define C_DATA_LEN          500
#define H_DATA_LEN          24

```

```

15 #define MAX_ONLINE_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 MakeAlphaStringPadded();
int MakeOriginalAlphaString();
int MakeNumberString();
int MakeZipNumberString();
void InitString();
void InitAddress();
void PaddString();

tpccldr.c

//=====
// File: TPCCLDR.C
//
// Microsoft TPC-C Kit Ver. 4.51
//
// 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 100000
#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
#define MAX_SQL_ERRORS 10

// Functions declarations
void HandleErrorDBC (SQLHDBC hdbc1);
long NURand();
void LoadItem();
void LoadWarehouse();
void Stock();
void District();
void LoadCustomer();
void CustomerBufInit();
void CustomerBufLoad();
void LoadCustomerTable();
void LoadHistoryTable();
void LoadOrders();
void OrdersBufInit();
void OrdersBufLoad();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void GetPermutation();
void CheckForCommit();
void CheckForCommit_Big();
void OpenConnections();
void BuildIndex();
void FormatDate ();

// Shared memory structures
typedef struct
{
    double
    ol;
    long
    ol_i_id;
    long
    ol_supply_w_id;
    short
    ol_quantity;
    double
    ol_amount;
    char
    ol_dist_info[DIST_INFO_LEN+1];

    char
    ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCTURE;

typedef struct
{
    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;
    ORDER_LINE_STRUCTURE
    o_ol[15];
} ORDERS_STRUCTURE;

typedef struct
{
    long
    c_id;
    short
    c_d_id;
    long
    c_w_id;

```



```

char      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;
char      c_balance[6];
double    c_ytd_payment;
short     c_payment_cnt;
short     c_delivery_cnt;
char      c_data[C_DATA_LEN+1];
double    h_amount;
char      h_data[H_DATA_LEN+1];
} CUSTOMER_STRUCT;

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

typedef struct
{
    long      time_start;
} LOADER_TIME_STRUCT;

// Global variables
char      szLastError[300];

HENV      henv;

HDBC      v_hdbc;
// for SQL

Server version verification
HDBC      i_hdbc1;
// for ITEM table
HDBC      w_hdbc1;
// for WAREHOUSE,
DISTRICT, STOCK
HDBC      c_hdbc1;
// for CUSTOMER
HDBC      c_hdbc2;
// for HISTORY

```

```

HDBC      o_hdbc1;
HDBC      o_hdbc2;
// for ORDERS
// for NEW-ORDER

HDBC      o_hdbc3;
// for ORDER-LINE

HSTMT     v_hstmt;
// for SQL Server version

verification
HSTMT     i_hstmt1;
HSTMT     w_hstmt1;
HSTMT     c_hstmt1, c_hstmt2;
HSTMT     o_hstmt1, o_hstmt2, o_hstmt3;

int      total_db_errors;

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

TPCCLDR_ARGS *aptr, args;

//=====
//
// Function name: main
//
//=====
int main(int argc, char **argv)
{
    DWORD
dwThreadId[MAX_MAIN_THREADS];
HANDLE
hThread[MAX_MAIN_THREADS];
FILE *fLoader;

```

```

char      buffer[255];
i;
for (i=0; i<MAX_MAIN_THREADS; i++)
    hThread[i] = NULL;

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

// process command line arguments
aptr = &args;
GetArgsLoader(argc, argv, aptr);

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

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

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

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

// open connections to SQL Server
OpenConnections();

// open file for loader results

```

```

fLoader = fopen(aptr->loader_res_file,
"w");

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

// start loading data
sprintf(buffer,"TPC-C load started for %ld
warehouses.\n",aptr->num_warehouses);
if (aptr->scale_down == 1)
{
sprintf(buffer,"SCALED
DOWN DATABASE.\n");
}

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

main_time_start = (TimeNow() / MILLI);

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

hThread[0] =
CreateThread(NULL,

0,

(LPTHREAD_START_ROUTINE) LoadItem,

NULL,

0,

&dwThreadID[0]);

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

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

hThread[1] =
CreateThread(NULL,

0,

(LPTHREAD_START_ROUTINE) LoadWarehouse,

```

```

NULL,

0,

&dwThreadID[1]);

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

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

hThread[2] =
CreateThread(NULL,

0,

(LPTHREAD_START_ROUTINE) LoadCustomer,

NULL,

0,

&dwThreadID[2]);

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

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

hThread[3] =
CreateThread(NULL,

0,

(LPTHREAD_START_ROUTINE) LoadOrders,

NULL,

0,

&dwThreadID[3]);

if (hThread[3] == NULL)

```

```

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

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

CloseHandle(hThread[i]);
hThread[i] =
NULL;
}

main_time_end = (TimeNow() / MILLI);

printf(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()
{
int i;
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(11);

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

//if build index before load

```

```

        if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
            BuildIndex("idxitmdl");

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

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

        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,
order (i_id), ROWS_PER_BATCH = 100000");
            rc = bcp_control(i_hdbc1,
BCPHINTS, (void*) bcphint);
            if (rc != SUCCEEDED)

                HandleErrorDBC(i_hdbc1);
        }

        i = 0;
        rc = bcp_bind(i_hdbc1, (BYTE *) &i_id,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(i_hdbc1);
        rc = bcp_bind(i_hdbc1, (BYTE *) i_name,
0, I_NAME_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(i_hdbc1);
        rc = bcp_bind(i_hdbc1, (BYTE *)
&i_price, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8,
++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(i_hdbc1);
        rc = bcp_bind(i_hdbc1, (BYTE *) i_data,
0, SQL_VARLEN_DATA, "", 1, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(i_hdbc1);
        rc = bcp_bind(i_hdbc1, (BYTE *)
&i_im_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
        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);

            MakeAlphaStringPadded(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("idxitmdl");
    }

    //=====
    //=====
    //=====
    //
    // Function : LoadWarehouse
    //
    // Loads WAREHOUSE table and loads Stock and
    // District as Warehouses are created
    //
    //=====
    //=====
    void LoadWarehouse()
    {
        int            i;
        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");
        rc = bcp_init(w_hdbc1, name, NULL,
err_log_path, DB_IN);

        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);

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

                HandleErrorDBC(w_hdbc1);
        }

        i = 0;
        rc = bcp_bind(w_hdbc1, (BYTE *)
&w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
&w_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8,
++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
&w_tax, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8,
++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
w_name, 0, W_NAME_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
w_street_1, 0, ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
w_street_2, 0, ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
w_city, 0, ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
w_state, 0, STATE_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) w_zip,
0, ZIP_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);

        time_start = (TimeNow() / MILLI);

        warehouse_rows_loaded = 0;

        for (w_id = (long)aptr-
>starting_warehouse; w_id <= aptr-
>num_warehouses; w_id++)
        {

```

```

        MakeAlphaStringPadded(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 != SUCCEEDED)

    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("idxwarc1");

stock_rows_loaded = 0;
district_rows_loaded = 0;

District();
Stock();
}

//=====
//
// Function : District
//
//=====
void District()
{
    int            i;
    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          bcp_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("idxdisc1");

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

strcpy(err_log_path,aptr->log_path);
strcat(err_log_path,"district.err");
rc = bcp_init(w_hdbc1, name, NULL,
err_log_path, DB_IN);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
{
    sprintf(bcp_hint, "tablock,
order (d_w_id, d_id), ROWS_PER_BATCH = %u",
(aptr->num_warehouses * 10));
    rc = bcp_control(w_hdbc1,
BCPHINTS, (void*) bcp_hint);
    if (rc != SUCCEEDED)

        HandleErrorDBC(w_hdbc1);
}

i = 0;
rc = bcp_bind(w_hdbc1, (BYTE *) &d_id,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *)
&d_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *)
&d_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8,
++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *)
&d_next_o_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *)
&d_tax, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8,
++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *)
d_name, 0, D_NAME_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *)
d_street_1, 0, ADDRESS_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)

    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *)
d_street_2, 0, ADDRESS_LEN, NULL, 0, 0, ++i);

```

```

if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *) d_city,
0, ADDRESS_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *)
d_state, 0, STATE_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *) d_zip,
0, ZIP_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

d_ytd = 30000.00;

d_next_o_id = orders_per_district+1;

time_start = (TimeNow() / MILLI);

for (w_id = aptr->starting_warehouse;
w_id <= aptr->num_warehouses; w_id++)
{
    d_w_id = w_id;

    for (d_id = 1; d_id <=
DISTRICT_PER_WAREHOUSE; d_id++)
    {

        MakeAlphaStringPadded(6,10,D_NAME_L
EN, 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 !=
SUCCEEDED)
    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("idxdisc1");

return;
}

//=====
//

```

```

// Function : Stock
//
//=====
void Stock()
{
    int          i;
    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("idxstkd");

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

    strcpy(err_log_path, aptr->log_path);
    strcat(err_log_path, "stock.err");
    rc = bcp_init(w_hdbc1, name, NULL,
err_log_path, DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);

    if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
    {
        sprintf(bcphint, "tablock,
order (s_i_id, s_w_id), ROWS_PER_BATCH = %u",
(aptr->num_warehouses * 100000));
        rc = bcp_control(w_hdbc1,
BCPHINTS, (void*) bcphint);
        if (rc != SUCCEEDED)

            HandleErrorDBC(w_hdbc1);
    }

    i = 0;
    rc = bcp_bind(w_hdbc1, (BYTE *)
&s_i_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);

    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);

    rc = bcp_bind(w_hdbc1, (BYTE *)
&s_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);

```

```

    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
&s_quantity, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
&s_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
&s_order_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
&s_remote_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
s_data, 0, SQL_VARLEN_DATA, "", 1, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_01, 0, S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_02, 0, S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_03, 0, S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_04, 0, S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_05, 0, S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_06, 0, S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_07, 0, S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_08, 0, S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_09, 0, S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_10, 0, S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        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 <= aptr-
>num_warehouses; s_w_id++)
        {
            (short)RandomNumber(10L, 100L);
            s_quantity =
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 !=
SUCCEEDED)
                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("idxstkd");

        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("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("idxhiscl");
        }

    // Initialize bulk copy
sprintf(name, "%s.%s", aptr->database,
"customer");
    strcpy(err_log_path_cust,aptr-
>log_path);
    strcat(err_log_path_cust,"customer.err");
    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,
order (c_w_id, c_d_id, c_id), ROWS_PER_BATCH =
%u", (aptr->num_warehouses * 30000));
        rc = bcp_control(c_hdbc1,
BCPHINTS, (void*) bcphint);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
    }

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

    rc = bcp_init(c_hdbc2, name, NULL,
"logs\\history.err", DB_IN);
    strcpy(err_log_path_hist,aptr-
>log_path);
    strcat(err_log_path_hist,"history.err");
    rc = bcp_init(c_hdbc2, name, NULL,
err_log_path_hist, DB_IN);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    sprintf(bcphint, "tablock");
    rc = bcp_control(c_hdbc2, BCPHINTS,
(void*) bcphint);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    customer_rows_loaded = 0;
    history_rows_loaded = 0;

    CustomerBufInit();

    customer_time_start.time_start =
(TimeNow() / MILLI);
    history_time_start.time_start =
(TimeNow() / MILLI);

    for (w_id = (long)aptr-
>starting_warehouse; w_id <= aptr-
>num_warehouses; w_id++)
    {
        for (d_id = 1; d_id <=
DISTRICT_PER_WAREHOUSE; d_id++)
        {
            CustomerBufLoad(d_id, w_id);

            // Start
            // Start
            customer table thread

            printf("...Loading customer table for:
d_id = %d, w_id = %d\n", d_id, w_id);

            hThread[0] =
CreateThread(NULL,

            0,

            (LPTHREAD_START_ROUTINE)
LoadCustomerTable,

            &customer_time_start,

            0,

```

```

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

            printf("...Loading history table for: d_id =
%d, w_id = %d\n", d_id, w_id);

            hThread[1] =
CreateThread(NULL,

            0,

            (LPTHREAD_START_ROUTINE)
LoadHistoryTable,

            &history_time_start,

            0,

            &dwThreadID[1]);

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

            WaitForSingleObject( hThread[0],
INFINITE );
            WaitForSingleObject( hThread[1],
INFINITE );

            if
(CloseHandle(hThread[0]) == FALSE)
            {
                printf("Error, failed in closing customer
thread handle with errno: %d\n", GetLastError());
            }

            if
(CloseHandle(hThread[1]) == FALSE)
            {
                printf("Error, failed in closing history
thread handle with errno: %d\n", GetLastError());
            }
        }

        // flush the bulk connection

```

```

rcint = bcp_done(c_hdbc1);
if (rcint < 0)
    HandleErrorDBC(c_hdbc1);

rcint = bcp_done(c_hdbc2);
if (rcint < 0)
    HandleErrorDBC(c_hdbc2);

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

// if build index after load...
if ((aptr->build_index == 1) && (aptr-
>index_order == 0))
{
    BuildIndex("idxcuscl");
// 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("idxhiscnc");
    }

// 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\"
> %snurand_load.log",

    aptr->server,
    aptr->user,
    aptr->password,
    aptr->database,
    LOADER_NURAND_C,
    aptr->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
//
// Fills shared buffer for HISTORY and CUSTOMER
//=====

```

```

//=====
//=====
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;

        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_N
URAND_C), c[i].c_last);

        MakeAlphaStringPadded(8,16,FIRST_NA
ME_LEN, c[i].c_first);

        c[i].c_id = i+1;
    }

    printf("...Loading customer buffer for:
d_id = %d, w_id = %d\n",
        d_id, w_id);

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

        customer_buf[i].c_ytd_payment = 10.0;

        customer_buf[i].c_payment_cnt = 1;

        customer_buf[i].c_delivery_cnt = 0;
        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;

strcpy(customer_buf[i].c_balance,"-
10.0");

MakeAlphaStringPadded(300, 500,
C_DATA_LEN, customer_buf[i].c_data);

// Generate HISTORY data

MakeAlphaStringPadded(12, 24,
H_DATA_LEN, customer_buf[i].h_data);
}

//=====
//
// Function : LoadCustomerTable
//
//=====
void LoadCustomerTable(LOADER_TIME_STRUCT
*customer_time_start)
{
    long          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;
    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;

    i = 0;
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_id,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *)
&c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0,
LAST_NAME_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_first,
0, FIRST_NAME_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0,
CREDIT_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5,
NULL, 0, SQLCHARACTER, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment,
0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *)
&c_delivery_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0,
STATE_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0,
ZIP_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0,
PHONE_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *)
&c_since, 0, C_SINCE_LEN, NULL, 0, SQLCHARACTER,
++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *)
c_middle, 0, MIDDLE_NAME_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0,
C_DATA_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        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;
        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 != SUCCEEDED)
            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;

    i = 0;
    rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *)
&c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *)
&c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *)
&h_date, 0, H_DATE_LEN, NULL, 0, SQLCHARACTER,
++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0,
H_DATA_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        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 != SUCCEEDED)

            HandleErrorDBC(c_hdbc2);

        history_rows_loaded++;
    }
}

```

```

        CheckForCommit(c_hdbc2,
c_hstmt2, history_rows_loaded, "history",
&hstmt_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("idxordc");

        BuildIndex("idxnodc");

        BuildIndex("idxodlc");
    }

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

    rc = bcp_init(o_hdbc1, name, NULL,
"log/orders.err", DB_IN);
    strcpy(err_log_path_ord,aptr-
>log_path);
    strcat(err_log_path_ord,"orders.err");
    rc = bcp_init(o_hdbc1, name, NULL,
err_log_path_ord, DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
}

```

```

        if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
            sprintf(bcphint, "tablock,
order (o_w_id, o_d_id, o_id), ROWS_PER_BATCH =
%u", (aptr->num_warehouses * 3000));
        rc = bcp_control(o_hdbc1,
BCPHINTS, (void*) bcphint);
        if (rc != SUCCEEDED)

            HandleErrorDBC(o_hdbc1);
    }

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

    rc = bcp_init(o_hdbc2, name, NULL,
"log/neword.err", DB_IN);
    strcpy(err_log_path_nord,aptr-
>log_path);
    strcat(err_log_path_nord,"neword.err");
    rc = bcp_init(o_hdbc2, name, NULL,
err_log_path_nord, DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);

    if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
    {
        sprintf(bcphint, "tablock,
order (no_w_id, no_d_id, no_o_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses
* 9000));
        rc = bcp_control(o_hdbc2,
BCPHINTS, (void*) bcphint);
        if (rc != SUCCEEDED)

            HandleErrorDBC(o_hdbc2);
    }

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

    rc = bcp_init(o_hdbc3, name, NULL,
"log/orderline.err", DB_IN);
    strcpy(err_log_path_ordl,aptr-
>log_path);
    strcat(err_log_path_ordl,"orderline.err");
    rc = bcp_init(o_hdbc3, name, NULL,
err_log_path_ordl, DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);

    if ((aptr->build_index == 1) && (aptr->index_order
== 1))
    {
        sprintf(bcphint, "tablock,
order (ol_w_id, ol_d_id, ol_o_id, ol_number),
ROWS_PER_BATCH = %u", (aptr->num_warehouses
* 300000));
        rc = bcp_control(o_hdbc3,
BCPHINTS, (void*) bcphint);
        if (rc != SUCCEEDED)

            HandleErrorDBC(o_hdbc3);
    }

    orders_rows_loaded = 0;
    new_order_rows_loaded = 0;
    order_line_rows_loaded = 0;

    OrdersBufInit();
}

```

```

        orders_time_start.time_start =
(TimeNow() / MILLI);
new_order_time_start.time_start =
(TimeNow() / MILLI);
order_line_time_start.time_start =
(TimeNow() / MILLI);

        for (w_id = (long)aptr-
>starting_warehouse; w_id <= aptr-
>num_warehouses; w_id++)
        {
                for (d_id = 1; d_id <=
DISTRICT_PER_WAREHOUSE; d_id++)
                {

                        OrdersBufLoad(d_id, w_id);

                                // start
parallel loading threads here...
                                // start Orders
table thread

                        printf("...Loading Order Table for: d_id =
%d, w_id = %d\n", d_id, w_id);

                                hThread[0] =
CreateThread(NULL,

                                0,

                                (LPTHREAD_START_ROUTINE)
LoadOrdersTable,

                                &orders_time_start,

                                0,

                                &dwThreadID[0]);

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

                                // start
NewOrder table thread

                        printf("...Loading New-Order Table for:
d_id = %d, w_id = %d\n", d_id, w_id);

                                hThread[1] =
CreateThread(NULL,

                                0,

                                (LPTHREAD_START_ROUTINE)
LoadNewOrderTable,

                                &new_order_time_start,

```

```

0,
&dwThreadID[1]);

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

                                // start Order-
Line table thread

                        printf("...Loading Order-Line Table for:
d_id = %d, w_id = %d\n", d_id, w_id);

                                hThread[2] =
CreateThread(NULL,

                                0,

                                (LPTHREAD_START_ROUTINE)
LoadOrderLineTable,

                                &order_line_time_start,

                                0,

                                &dwThreadID[2]);

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

                                WaitForSingleObject( hThread[0],
INFINITE );

                                WaitForSingleObject( hThread[1],
INFINITE );

                                WaitForSingleObject( hThread[2],
INFINITE );

                                if
(CloseHandle(hThread[0]) == FALSE)
                                {
                                        printf("Error, failed in closing Orders
thread handle with errno: %d\n", GetLastError());
                                }

                                if
(CloseHandle(hThread[1]) == FALSE)
                                {

```

```

                                printf("Error, failed in closing NewOrder
thread handle with errno: %d\n", GetLastError());
                                }
                                if
(CloseHandle(hThread[2]) == FALSE)
                                {
                                        printf("Error, failed in closing OrderLine
thread handle with errno: %d\n", GetLastError());
                                }
                                }

                                printf("Finished loading orders.\n");

                                return;
}

//=====
// Function : OrdersBufInit
//
// Clears shared buffer for ORDERS, NEWORDER, and
ORDERLINE
//=====
void OrdersBufInit()
{
        int i;
        int j;

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

                orders_buf[i].o_ol_cnt = 0;
                orders_buf[i].o_all_local =
0;

                for (j=0;j<=14;j++)
                {

                        orders_buf[i].o_ol[j].ol = 0;

                        orders_buf[i].o_ol[j].ol_i_id = 0;

                        orders_buf[i].o_ol[j].ol_supply_w_id = 0;

                        orders_buf[i].o_ol[j].ol_quantity = 0;

                        orders_buf[i].o_ol[j].ol_amount = 0;

                        strcpy(orders_buf[i].o_ol[j].ol_dist_info,"
");
                }
        }

//=====
// Function : OrdersBufLoad
//

```

```

// Fills shared buffer for ORDERS, NEWORDER, and
ORDERLINE
//
//=====
//=====
void OrdersBufLoad(short d_id, long w_id)
{
    int cust[ORDERS_PER_DISTRICT+1];
    long o_id;
    long ol;

    printf("...Loading Order Buffer for: d_id
= %d, w_id = %d\n", d_id, w_id);

    GetPermutation(cust,
orders_per_district);

    for
(o_id=0;o_id<orders_per_district;o_id++)
    {
        // Generate ORDER and
NEW-ORDER data
orders_buf[o_id].o_d_id =
d_id;
orders_buf[o_id].o_w_id =
w_id;
orders_buf[o_id].o_id =
o_id+1;
orders_buf[o_id].o_c_id =
cust[o_id+1];
orders_buf[o_id].o_ol_cnt
= (short)RandomNumber(5L, 15L);

        if (o_id < first_new_order)
        {
            orders_buf[o_id].o_carrier_id =
(short)RandomNumber(1L, 10L);

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

            orders_buf[o_id].o_all_local = 1;
        }

        for (ol=0;
ol<orders_buf[o_id].o_ol_cnt; ol++)
        {
            orders_buf[o_id].o_ol[ol].ol = ol+1;

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

            orders_buf[o_id].o_ol[ol].ol_supply_w_id
= w_id;

            orders_buf[o_id].o_ol[ol].ol_quantity =
5;

            MakeAlphaString(24, 24,
OL_DIST_INFO_LEN,
&orders_buf[o_id].o_ol[ol].ol_dist_info);

            // Generate
ORDER-LINE data

```

```

if (o_id <
first_new_order)
{
    orders_buf[o_id].o_ol[ol].ol_amount = 0;

    // Added to insure ol_delivery_d set
properly during load
    FormatDate(&orders_buf[o_id].o_ol[ol].o
l_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_delive
ry_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
    i = 0;
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_id,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *)
&o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);

```

```

rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *)
&o_entry_d, 0, O_ENTRY_D_LEN, NULL, 0,
SQLCHARACTER, ++);
    if (rc != SUCCEEDED)
        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 != SUCCEEDED)
            HandleErrorDBC(o_hdbc1);

            orders_rows_loaded++;
            CheckForCommit(o_hdbc1,
o_hstmt1, orders_rows_loaded, "orders",
&orders_time_start->time_start);
        }

        if ((o_w_id == apr->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 ((apr->build_index ==
1) && (apr->index_order == 0))
                    BuildIndex("idxordc1");

```

```

// build non-clustered index
if (aptr->build_index == 1)

    BuildIndex("idxordnc");
}

//=====
//
// Function : LoadNewOrderTable
//
//=====
void LoadNewOrderTable(LOADER_TIME_STRUCT
*new_order_time_start)
{
    long          i;
    long          o_id;
    short         o_d_id;
    long          o_w_id;
    RETCODE       rc;
    DBINT         rcint;

    // Bind NEW-ORDER data
    i = 0;
    rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);
    rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);
    rc = bcp_bind(o_hdbc2, (BYTE *)
&o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);

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

    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("idxnodc");
    }
}

//=====
//
// 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_delivery_d[OL_DELIVERY_D_LEN+1];
    RETCODE       rc;
    DBINT         rcint;

    // bind ORDER-LINE data
    i = 0;
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *)
&o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *)
&ol_delivery_d, 0, OL_DELIVERY_D_LEN, NULL, 0,
SQL_CHARACTER, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);

```

```

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *)
&ol_supply_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0,
DIST_INFO_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);

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

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

            rc =
            bcp_sendrow(o_hdbc3);
            if (rc !=
SUCCEEDED)
                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);
        }

        if ((o_w_id == aptr->num_warehouses)
&& (o_d_id == 10))
        {
            rcint =
            bcp_done(o_hdbc3);
            if (rcint < 0)

```

```

        HandleErrorDBC(o_hdbc3);

        SQLFreeStmt(o_hstmt3,
SQL_DROP);
        SQLDisconnect(o_hdbc3);
        SQLFreeConnect(o_hdbc3);

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

            BuildIndex("idxodcl");
    }

//=====
//=====
//
// Function : GetPermutation
//
//=====
void GetPermutation(int perm[], int n)
{
    int i, r, t;

    for (i=1;i<=n;i++)
        perm[i] = i;

    for (i=1;i<=n;i++)
    {
        r = RandomNumber(i,n);
        t = perm[i];
        perm[i] = perm[r];
        perm[r] = t;
    }
}

//=====
//=====
//
// Function : CheckForCommit
//
//=====
void CheckForCommit(HDBC hdbc,
                    HSTMT hstmt,
                    long rows_loaded,
                    char *table_name,
                    long *time_start)
{
    long    time_end, time_diff;

    if ( !(rows_loaded % aptr->batch) )
    {
        time_end = (TimeNow() /
MILLI);
        time_diff = time_end -
*time_start;

        printf("> Loaded %ld rows
into %s in %ld sec - Total = %d (%.2f rps)\n",
                    aptr->batch,
                    table_name,
                    time_diff,
                    rows_loaded,
                    (float) aptr->batch / (time_diff ?
time_diff : 1L));

        *time_start = time_end;
    }

    return;
}

//=====
//=====
//
// Function : OpenConnections
//
//=====
void OpenConnections()
{
    RETCODE    rc;

```

```

        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 = %d (%.2f rps)\n",
                    aptr->batch,
                    table_name,
                    time_diff,
                    rows_loaded,
                    (float) aptr->batch / (time_diff ?
time_diff : 1L));

        *time_start = time_end;
    }

    return;
}

//=====
//=====
//
// Function : OpenConnections
//
//=====
void OpenConnections()
{
    RETCODE    rc;

```

```

char
szDriverString[300];
szDriverStringOut[1024];
SQLSMALLINT
cbDriverStringOut;

SQLAllocHandle(SQL_HANDLE_ENV,
SQL_NULL_HANDLE, &henv );

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

SQLAllocHandle(SQL_HANDLE_DBC,
henv , &i_hdbc1);
SQLAllocHandle(SQL_HANDLE_DBC,
henv , &w_hdbc1);
SQLAllocHandle(SQL_HANDLE_DBC,
henv , &c_hdbc1);
SQLAllocHandle(SQL_HANDLE_DBC,
henv , &c_hdbc2);
SQLAllocHandle(SQL_HANDLE_DBC,
henv , &o_hdbc1);
SQLAllocHandle(SQL_HANDLE_DBC,
henv , &o_hdbc2);
SQLAllocHandle(SQL_HANDLE_DBC,
henv , &o_hdbc3);

SQLSetConnectAttr(i_hdbc1,
SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );

SQLSetConnectAttr(w_hdbc1,
SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
SQLSetConnectAttr(c_hdbc1,
SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
SQLSetConnectAttr(c_hdbc2,
SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
SQLSetConnectAttr(o_hdbc1,
SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
SQLSetConnectAttr(o_hdbc2,
SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
SQLSetConnectAttr(o_hdbc3,
SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );

// Open connections to SQL Server
// Connection 1
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=
%s" ,

        aptr->server,

        aptr->user,

        aptr->password,

        aptr->database );

    rc = SQLSetConnectOption ( i_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    rc = SQLDriverConnect ( i_hdbc1,

        NULL,

```

```

(SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,
(SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT
);
if ( (rc != SUCCEEDED) &&
    (rc !=
SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(i_hdbc1);
    printf("TPC-C Loader
aborted!\n");
    exit(9);
}

// Connection 2
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=
%s" ,

        aprt->server,
        aprt->user,
        aprt->password,
        aprt->database );

rc = SQLSetConnectOption (w_hdbc1,
SQL_PACKET_SIZE, aprt->pack_size);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = SQLDriverConnect ( w_hdbc1,
        NULL,
        (SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,
        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEEDED) &&
    (rc !=
SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(w_hdbc1);
    printf("TPC-C Loader
aborted!\n");
    exit(9);
}

```

```

        sprintf(szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=
%s" ,

        aprt->server,
        aprt->user,
        aprt->password,
        aprt->database );

rc = SQLSetConnectOption (c_hdbc1,
SQL_PACKET_SIZE, aprt->pack_size);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);

rc = SQLDriverConnect ( c_hdbc1,
        NULL,
        (SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,
        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEEDED) &&
    (rc !=
SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(c_hdbc1);
    printf("TPC-C Loader
aborted!\n");
    exit(9);
}

// Connection 4
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=
%s" ,

        aprt->server,
        aprt->user,
        aprt->password,
        aprt->database );

rc = SQLSetConnectOption (c_hdbc2,
SQL_PACKET_SIZE, aprt->pack_size);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc2);

rc = SQLDriverConnect ( c_hdbc2,
        NULL,
        (SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,

```

```

        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEEDED) &&
    (rc !=
SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(c_hdbc2);
    printf("TPC-C Loader
aborted!\n");
    exit(9);
}

// Connection 5
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=
%s" ,

        aprt->server,
        aprt->user,
        aprt->password,
        aprt->database );

rc = SQLSetConnectOption (o_hdbc1,
SQL_PACKET_SIZE, aprt->pack_size);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc1);

rc = SQLDriverConnect ( o_hdbc1,
        NULL,
        (SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,
        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEEDED) &&
    (rc !=
SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(o_hdbc1);
    printf("TPC-C Loader
aborted!\n");
    exit(9);
}

// Connection 6
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=
%s" ,

```

```

aptr->server,
aptr->user,
aptr->password,
aptr->database );

rc = SQLSetConnectOption (o_hdbc2,
SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc2);

rc = SQLDriverConnect ( o_hdbc2,
                        NULL,
                        (SQLCHAR*)&szDriverString[0] ,
                        SQL_NTS,
                        (SQLCHAR*)&szDriverStringOut[0],
                        sizeof(szDriverStringOut),
                        &cbDriverStringOut,
                        SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEED) &&
      (rc !=
SQL_SUCCESS_WITH_INFO) )
    HandleErrorDBC(o_hdbc2);
    printf("TPC-C Loader
aborted!\n");
    exit(9);
}

//=====
//
// 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
Msg[SQL_MAX_MESSAGE_LENGTH];
    SQLLEN
NativeError;

```

```

        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
    if ( (rc != SUCCEED) &&
          (rc !=
SQL_SUCCESS_WITH_INFO) )
    {
        HandleErrorDBC(o_hdbc3);
        printf("TPC-C Loader
aborted!\n");
        exit(9);
    }
}

//=====
//
// 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
Msg[SQL_MAX_MESSAGE_LENGTH];
    SQLLEN
NativeError;

```

```

        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 )
        {
            printf( szLastError , "%s" ,
Msg );

            _strtime(timebuf);
            _strdate(datebuf);

            printf( "[%s : %s]
%s\n==>SQLState: %s\n" , datebuf, timebuf,
szLastError, SqlState);

            strcpy(err_log_path,aptr-
>log_path);

            strcat(err_log_path,"tpccldr.err");
            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];
    SQLLEN
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 (total_db_errors >=
MAX_SQL_ERRORS)
    {
        printf(">>>> Maximum SQL errors of
%d exceeded. Terminating
TPCCldr.<<<<<<\n",total_db_errors);
        exit(9);
    }
    total_db_errors++;
    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");
    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;
}

```

### time.c

```

// File: TIME.C
//
// Microsoft TPC-C Kit Ver. 4.62
//
// Copyright Microsoft, 1996, 1997, 1998,
1999, 2000, 2001, 2002, 2005
// 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("[%d]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
//
// Microsoft TPC-C Kit Ver. 4.51
//
// 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("[%d]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("[%d]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 <%d> 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[] =
"0123456789ABCDEFGHIJKLMN0PQRSTUVWXYZabcde
fghijklmnopqrstuvwxyz";
    static int chArrayMax =
61;

    for (i=0; i<len; i++)
        str[i] =
chArray[RandomNumber(0,chArrayMax)];
    str[len] = 0;

    return len;
}

int MakeAlphaStringPadded( int minLen, int maxLen,
int padLen, char *str)
{
    int len;
    int i;
    char cc = 'a';
    static char chArray[] =
"0123456789ABCDEFGHIJKLMN0PQRSTUVWXYZabcde
fghijklmnopqrstuvwxyz";
    static int chArrayMax =
61;

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

    len= RandomNumber(minLen, maxLen);

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

    return padLen;
}

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

int MakeOriginalAlphaString(int x,
int y,
int z,
char *str,
int percent)
{
#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 < 8)
    {
        printf("MakeOriginalAlphaString: string
length must be >= 8\n");
        exit(-1);
    }

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

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

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

    return len;
}

//=====
// Function name: MakeNumberString
//=====

int MakeNumberString(int x, int y, int z, char *str)
{
    char tmp[16];

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

    memset(str, '0', 16);
    itoa(RandomNumber(0, 999999999), tmp,
10);
    memcpy(str, tmp, strlen(tmp));

    itoa(RandomNumber(0, 999999999), tmp,
10);
    memcpy(str+8, tmp, strlen(tmp));

    str[16] = 0;
}

```

```

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

    len= RandomNumber(x, y);

    for (i=0; i<len; i++)
        str[i] =
chArray[RandomNumber(0,chArrayMax)];
    str[len] = 0;

    return len;
}

int MakeAlphaStringPadded( int minLen, int maxLen,
int padLen, char *str)
{
    int len;
    int i;
    char cc = 'a';
    static char chArray[] =
"0123456789ABCDEFGHIJKLMN0PQRSTUVWXYZabcde
fghijklmnopqrstuvwxyz";
    static int chArrayMax =
61;

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

    len= RandomNumber(minLen, maxLen);

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

    return padLen;
}

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

int MakeOriginalAlphaString(int x,
int y,
int z,
char *str,
int percent)
{
#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 < 8)
    {
        printf("MakeOriginalAlphaString: string
length must be >= 8\n");
        exit(-1);
    }

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

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

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

    return len;
}

//=====
// Function name: MakeNumberString
//=====

int MakeNumberString(int x, int y, int z, char *str)
{
    char tmp[16];

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

    memset(str, '0', 16);
    itoa(RandomNumber(0, 999999999), tmp,
10);
    memcpy(str, tmp, strlen(tmp));

    itoa(RandomNumber(0, 999999999), tmp,
10);
    memcpy(str+8, tmp, strlen(tmp));

    str[16] = 0;
}

```

```

#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 < 8)
    {
        printf("MakeOriginalAlphaString: string
length must be >= 8\n");
        exit(-1);
    }

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

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

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

    return len;
}

//=====
// Function name: MakeNumberString
//=====

int MakeNumberString(int x, int y, int z, char *str)
{
    char tmp[16];

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

    memset(str, '0', 16);
    itoa(RandomNumber(0, 999999999), tmp,
10);
    memcpy(str, tmp, strlen(tmp));

    itoa(RandomNumber(0, 999999999), 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;
}

```

```

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. 4.62
// Copyright Microsoft, 1996, 1997, 1998,
1999, 2000, 2001, 2002, 2005
// 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
%d\n", Seed, val);
#endif

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

    Seed = val;
}

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

long irand()

```

```

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

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

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

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

    return( Seed );
}

/*****
*****
*
* drand - returns a double pseudo random number
between 0.0 and 1.0.
* See irand.
*
*****
*****/
double drand()
{
#ifdef DEBUG
    printf("[%d]DBG: Entering drand()...\n", (int)
GetCurrentThreadId());
#endif

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

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

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

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

    upper++;

    if ( upper <= lower )
        rand_num = upper;
    else

```

```

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

#ifdef DEBUG
    printf("[%d]DBG: RandomNumber between %ld &
%ld ==> %ld\n",

        (int) GetCurrentThreadId(), lower, upper,
rand_num);
#endif

    return rand_num;
}

#if 0
//Original code pgd 08/13/96

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

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

    upper++;

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

#ifdef DEBUG
    printf("[%d]DBG: RandomNumber between %ld &
%ld ==> %ld\n",

        (int) GetCurrentThreadId(), lower, upper,
rand_num);
#endif

    return rand_num;
}

#endif

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

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

```

```

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

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

    return rand_num;
}

}

getargs.c

// File: GETARGS.C
//
// Microsoft TPC-C Kit Ver. 4.51
//
// Copyright Microsoft, 1996, 1997, 1998,
1999, 2000, 2001, 2002, 2003
// Purpose: Source file for command
line processing

// Includes
#include "tpcc.h"

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

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

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

    #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
= DEFLDAPACKSIZE;
pargs->starting_warehouse
= DEF_STARTING_WAREHOUSE;
pargs->build_index
= BUILD_INDEX;
pargs->index_order
= INDEX_ORDER;
pargs->index_script_path =
INDEX_SCRIPT_PATH;
pargs->scale_down
= SCALE_DOWN;

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

for (i = 1; i < argc; ++i)
{
    if (argv[i][0] != '-' &&
argv[i][0] != '/')
        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':
pargs->batch = atol(ptr+2);
        break;

```

```

        case 'W':
pargs->num_warehouses = atol(ptr+2);
        break;

        case 's':
pargs->starting_warehouse =
atol(ptr+2);
        break;

        case 't':
{
pargs->tables_all = FALSE;

if (strcmp(ptr+2,"item") == 0)
    pargs->table_item = TRUE;

else if (strcmp(ptr+2,"warehouse") ==
0)
    pargs->table_warehouse =
TRUE;

else if (strcmp(ptr+2,"customer") == 0)
    pargs->table_customer =
TRUE;

else if (strcmp(ptr+2,"orders") == 0)
    pargs->table_orders =
TRUE;

else
{
    printf("\nUnrecognized command");
    GetArgsLoaderUsage();
    exit(1);
}

        break;

        case 'f':
pargs->loader_res_file = ptr+2;
        break;

        case 'L':
pargs->log_path = ptr+2;
        break;

        case 'p':
pargs->pack_size = atol(ptr+2);
        break;

```

```

        case 'i':
pargs->build_index = atol(ptr+2);
        break;

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

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

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

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

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

return;
}

//=====
//
// Function name: GetArgsLoaderUsage
//
//=====
void GetArgsLoaderUsage()
{
#ifdef DEBUG
    printf("[%d]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("-L Loader BCP Log Path
%s\n", LOADER_LOG_PATH);
printf("-f Loader Results Output Filename
%s\n", LOADER_RES_FILE);
printf("-s Starting Warehouse
%ld\n", (long) DEF_STARTING_WAREHOUSE);
printf("-i Build Option (data = 0, data
and index = 1) %ld\n", (long) BUILD_INDEX);
printf("-o Cluster Index Build Order
(before = 1, after = 0) %ld\n", (long)
INDEX_ORDER);
printf("-c Build Scaled Database (normal
= 0, tiny = 1) %ld\n", (long) SCALE_DOWN);
printf("-d Index Script Path
%s\n", INDEX_SCRIPT_PATH);
printf("-t Table to Load
all tables \n");
printf(" [item|warehouse|customer|orders]\n");
printf(" Notes: \n");
printf(" - the '-t' parameter may be included
multiple times to \n");
printf(" specify multiple tables to be loaded
\n");
printf(" - 'item' loads ITEM table \n");
printf(" - 'warehouse' loads WAREHOUSE,
DISTRICT, and STOCK tables \n");
printf(" - 'customer' loads CUSTOMER and
HISTORY tables \n");
printf(" - 'orders' load NEW-ORDER, ORDERS,
ORDER-LINE tables \n");

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

exit(0);
}

```

## B.1 Database Options

### DBOPT1 . SQL

```

-----
-- File: DBOPT1.SQL --
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005 --
-- Sets database options for load
--
-----
USE master
GO

```

```

ALTER DATABASE tpcc SET RECOVERY BULK_LOGGED
GO

EXEC sp_dboption tpcc,'trunc. log on chkpt.',TRUE
GO

ALTER DATABASE tpcc SET TORN_PAGE_DETECTION
OFF
GO

ALTER DATABASE tpcc SET PAGE_VERIFY NONE
GO

USE tpcc
GO

CHECKPOINT
GO

```

### DBOPT2 . SQL

```

-----
-- File: DBOPT2.SQL --
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005 --
-- Sets database options after load
--
-----
ALTER DATABASE tpcc SET RECOVERY FULL
GO

USE tpcc
GO

CHECKPOINT
GO

sp_configure 'allow updates',1
GO

RECONFIGURE WITH OVERRIDE
GO

DECLARE @msg varchar(50)

-----
-- OPTIONS FOR SQL SERVER 2000 --
-- Set option values for user-defined indexes --
-----

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

EXEC sp_indexoption 'customer',
'DisAllowPageLocks', TRUE

```

```

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

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

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

sp_configure 'allow updates',0
GO

RECONFIGURE WITH OVERRIDE
GO

EXEC sp_dboption tpcc, 'auto update statistics',
FALSE
EXEC sp_dboption tpcc, 'auto create statistics',
FALSE
GO

DECLARE @db_id int,
@tbl_id int

SET @db_id = DB_ID('tpcc')
SET @tbl_id = OBJECT_ID('tpcc..warehouse')
DBCC PINTABLE (@db_id, @tbl_id)

SET @tbl_id = OBJECT_ID('tpcc..district')
DBCC PINTABLE (@db_id, @tbl_id)

SET @tbl_id = OBJECT_ID('tpcc..new_order')
DBCC PINTABLE (@db_id, @tbl_id)

SET @tbl_id = OBJECT_ID('tpcc..item')
DBCC PINTABLE (@db_id, @tbl_id)
GO

```

## B.2 Table definitions

### Createdb.sql

```

-----
--
-- File: CREATEDB.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.60
-- Copyright Microsoft, 2005
--
-----

SET ANSI_NULL_DFLT_OFF ON
GO

USE master
GO

-----
-- Create temporary table for timing
-----

IF EXISTS( SELECT name FROM sysobjects WHERE
name = 'tpcc_timer' )
    DROP TABLE tpcc_timer
GO

CREATE TABLE tpcc_timer
    (start_date CHAR(30),
    end_date CHAR(30))
GO

INSERT INTO tpcc_timer VALUES(0,0)
GO

-----
-- Store starting time
-----

UPDATE tpcc_timer
SET start_date = (SELECT
CONVERT(CHAR(30), GETDATE(), 21))
GO

-----
-- create main database files
-----

CREATE DATABASE tpcc
ON PRIMARY
(
    NAME = tpccRoot,
    FILENAME = "g:\tpcc25000.mdf",
    SIZE = 8MB,
    FILEGROWTH = 0),

FILEGROUP cs_fg
    (NAME=cs1, FILENAME="g:\mnt\cs\1",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs2, FILENAME="g:\mnt\cs\2",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs3, FILENAME="g:\mnt\cs\3",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs4, FILENAME="g:\mnt\cs\4",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs5, FILENAME="g:\mnt\cs\5",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs6, FILENAME="g:\mnt\cs\6",
    SIZE=86000MB,FILEGROWTH=0),

```

```

    (NAME=cs7, FILENAME="g:\mnt\cs\7",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs8, FILENAME="g:\mnt\cs\8",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs9, FILENAME="g:\mnt\cs\9",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs10,
    FILENAME="g:\mnt\cs\10",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs11,
    FILENAME="g:\mnt\cs\11",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs12,
    FILENAME="g:\mnt\cs\12",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs13,
    FILENAME="g:\mnt\cs\13",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs14,
    FILENAME="g:\mnt\cs\14",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs15,
    FILENAME="g:\mnt\cs\15",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs16,
    FILENAME="g:\mnt\cs\16",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs17,
    FILENAME="g:\mnt\cs\17",
    SIZE=86000MB,FILEGROWTH=0),
    (NAME=cs18,
    FILENAME="g:\mnt\cs\18",
    SIZE=86000MB,FILEGROWTH=0),

FILEGROUP misc_fg
    (NAME=misc1,
    FILENAME="g:\mnt\misc\1",SIZE=48100MB,FILEGRO
    WTH=0),
    (NAME=misc2,
    FILENAME="g:\mnt\misc\2",SIZE=48100MB,FILEGRO
    WTH=0),
    (NAME=misc3,
    FILENAME="g:\mnt\misc\3",SIZE=48100MB,FILEGRO
    WTH=0),
    (NAME=misc4,
    FILENAME="g:\mnt\misc\4",SIZE=48100MB,FILEGRO
    WTH=0),
    (NAME=misc5,
    FILENAME="g:\mnt\misc\5",SIZE=48100MB,FILEGRO
    WTH=0),
    (NAME=misc6,
    FILENAME="g:\mnt\misc\6",SIZE=48100MB,FILEGRO
    WTH=0),
    (NAME=misc7,
    FILENAME="g:\mnt\misc\7",SIZE=48100MB,FILEGRO
    WTH=0),
    (NAME=misc8,
    FILENAME="g:\mnt\misc\8",SIZE=48100MB,FILEGRO
    WTH=0),
    (NAME=misc9,
    FILENAME="g:\mnt\misc\9",SIZE=48100MB,FILEGRO
    WTH=0),
    (NAME=misc10,
    FILENAME="g:\mnt\misc\10",SIZE=48100MB,FILEGR
    OWTH=0),
    (NAME=misc11,
    FILENAME="g:\mnt\misc\11",SIZE=48100MB,FILEGR
    OWTH=0),
    (NAME=misc12,
    FILENAME="g:\mnt\misc\12",SIZE=48100MB,FILEGR
    OWTH=0),
    (NAME=misc13,
    FILENAME="g:\mnt\misc\13",SIZE=48100MB,FILEGR
    OWTH=0),

```

```

    (NAME=misc14,
    FILENAME="g:\mnt\misc\14",SIZE=48100MB,FILEGR
    OWTH=0),
    (NAME=misc15,
    FILENAME="g:\mnt\misc\15",SIZE=48100MB,FILEGR
    OWTH=0),
    (NAME=misc16,
    FILENAME="g:\mnt\misc\16",SIZE=48100MB,FILEGR
    OWTH=0),
    (NAME=misc17,
    FILENAME="g:\mnt\misc\17",SIZE=48100MB,FILEGR
    OWTH=0),
    (NAME=misc18,
    FILENAME="g:\mnt\misc\18",SIZE=48100MB,FILEGR
    OWTH=0)

LOG ON
(
    NAME = tpcc_log,
    FILENAME = "L:",
    SIZE = 20000MB,
    FILEGROWTH = 0)

COLLATE Latin1_General_BIN
GO

-----
-- Store ending time
-----

UPDATE tpcc_timer
SET end_date = (SELECT
CONVERT(CHAR(30), GETDATE(), 21))
GO

SELECT DATEDIFF(second,(SELECT start_date FROM
tpcc_timer),(SELECT end_date FROM tpcc_timer))
GO

-----
-- remove temporary table
-----

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'tpcc_timer' )
    DROP TABLE tpcc_timer
GO

```

### tables.sql

```

-----
--
-- File: TABLES.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.62
-- Copyright Microsoft, 2005
-- Creates TPC-C tables
--
-----

SET ANSI_NULL_DFLT_OFF ON

GO

USE tpcc
GO

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

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

```

```

go
if exists ( select name from sysobjects where name =
'district' )
    drop table district
go
if exists ( select name from sysobjects where name =
'customer' )
    drop table customer
go
if exists ( select name from sysobjects where name =
'history' )
    drop table history
go
if exists ( select name from sysobjects where name =
'new_order' )
    drop table new_order
go
if exists ( select name from sysobjects where name =
'orders' )
    drop table orders
go
if exists ( select name from sysobjects where name =
'order_line' )
    drop table order_line
go
if exists ( select name from sysobjects where name =
'item' )
    drop table item
go
if exists ( select name from sysobjects where name =
'stock' )
    drop table stock
go
-----
-- Create new tables
-----
create table warehouse
(
    w_id          int,
    w_ytd         money,
    w_tax         smallmoney,
    w_name        char(10),
    w_street_1    char(20),
    w_street_2    char(20),
    w_city        char(20),
    w_state       char(2),
    w_zip         char(9)
) on misc_fg
go
create table district
(
    d_id          tinyint,
    d_w_id        int,
    d_ytd         money,
    d_next_o_id   int,
    d_tax         smallmoney,
    d_name        char(10),
    d_street_1    char(20),
    d_street_2    char(20),
    d_city        char(20),
    d_state       char(2),
    d_zip         char(9)
) on misc_fg
go
create table customer
(
    c_id          int,
    c_d_id        tinyint,
    c_w_id        int,
    c_discount    smallmoney,
    c_credit_lim  money,
    c_last        char(16),

```

```

    c_credit      char(16),
    c_balance     money,
    c_ytd_payment money,
    c_payment_cnt smallint,
    c_delivery_cnt smallint,
    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_middle      char(2),
    c_data        char(500)
) on cs_fg
go
-- Use the following table option if using c_data
varchar(max)
-- sp_tableoption 'customer',large value types out of
row',1'
-- go
create table history
(
    h_c_id        int,
    h_c_d_id      tinyint,
    h_c_w_id      int,
    h_d_id        tinyint,
    h_w_id        int,
    h_date        datetime,
    h_amount      smallmoney,
    h_data        char(24)
) on misc_fg
go
create table new_order
(
    no_o_id       int,
    no_d_id       tinyint,
    no_w_id       int
) on misc_fg
go
create table orders
(
    o_id          int,
    o_d_id        tinyint,
    o_w_id        int,
    o_c_id        int,
    o_carrier_id  tinyint,
    o_ol_cnt      tinyint,
    o_all_local   tinyint,
    o_entry_d     datetime
) on misc_fg
go
create table order_line
(
    ol_o_id       int,
    ol_d_id       tinyint,
    ol_w_id       int,
    ol_number     tinyint,
    ol_i_id       int,
    ol_delivery_d datetime,
    ol_amount     smallmoney,
    ol_supply_w_id int,
    ol_quantity   smallint,
    ol_dist_info  char(24)
) on misc_fg
go

```

```

create table item
(
    i_id          int,
    i_name        char(24),
    i_price       smallmoney,
    i_data        char(50),
    i_im_id       int
) on misc_fg
go
create table stock
(
    s_i_id        int,
    s_w_id        int,
    s_quantity    smallint,
    s_ytd         int,
    s_order_cnt   smallint,
    s_remote_cnt  smallint,
    s_data        char(50),
    s_dist_01     char(24),
    s_dist_02     char(24),
    s_dist_03     char(24),
    s_dist_04     char(24),
    s_dist_05     char(24),
    s_dist_06     char(24),
    s_dist_07     char(24),
    s_dist_08     char(24),
    s_dist_09     char(24),
    s_dist_10     char(24)
) on cs_fg
go

```

### Clustered Index Creation Scripts

```

-----
--
-- File: IDXCUSCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.62
--
-- Copyright Microsoft, 2005
--
-- Creates clustered index on customer table
-----
USE tpcc
GO
DECLARE @startdate DATETIME,
        @enddate   DATETIME
SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)
IF EXISTS ( SELECT name FROM sysindexes WHERE
name = 'customer_c1' )
    DROP INDEX customer.customer_c1
CREATE UNIQUE CLUSTERED INDEX customer_c1 ON
customer(c_w_id, c_d_id, c_id)
ON cs_fg
SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO
-----

```

```

--
-- File: IDXDISCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.62
--
-- Copyright Microsoft, 2005
--
-- Creates clustered index on district table
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE
name = 'district_c1' )
DROP INDEX district.district_c1

CREATE UNIQUE CLUSTERED INDEX district_c1 ON
district(d_w_id, d_id)
WITH FILLFACTOR=100 ON misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

-----
-- File: IDXHISCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.62
--
-- Copyright Microsoft, 2005
--
-- Creates clustered index on history table
--
-- CAUTION: This index is only beneficial for
systems --
-- CAUTION: with 8 or more processors.
--
-- CAUTION: It may negatively impact
performance on --
-- CAUTION: systems with less than 8
processors. --
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE
name = 'history_c1' )
DROP INDEX history.history_c1

CREATE UNIQUE CLUSTERED INDEX history_c1 ON
history(h_c_w_id, h_date, h_c_d_id, h_c_id,
h_amount)
ON misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',

```

```

SELECT CONVERT(VARCHAR(30),@startdate,21)
DATEDIFF(second, @startdate, @enddate)
GO

-----
-- File: IDXITMCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.62
--
-- Copyright Microsoft, 2005
--
-- Creates clustered index on item table
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE
name = 'item_c1' )
DROP INDEX item.item_c1

CREATE UNIQUE CLUSTERED INDEX item_c1 ON
item(i_id)
ON misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

-----
-- File: IDXNODCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.62
--
-- Copyright Microsoft, 2005
--
-- Creates clustered index on new-order table
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE
name = 'new_order_c1' )
DROP INDEX new_order.new_order_c1

CREATE UNIQUE CLUSTERED INDEX new_order_c1
ON new_order(no_w_id, no_d_id, no_o_id)

ON misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',

```

```

SELECT CONVERT(VARCHAR(30),@startdate,21)
DATEDIFF(second, @startdate, @enddate)
GO

-----
-- File: IDXODLCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.62
--
-- Copyright Microsoft, 2005
--
-- Creates clustered index on order-line table
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE
name = 'order_line_c1' )
DROP INDEX order_line.order_line_c1

CREATE UNIQUE CLUSTERED INDEX order_line_c1 ON
order_line(ol_w_id, ol_d_id, ol_o_id, ol_number)
ON misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

-----
-- File: IDXORDCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.62
--
-- Copyright Microsoft, 2005
--
-- Creates clustered index on orders table
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE
name = 'orders_c1' )
DROP INDEX orders.orders_c1

CREATE UNIQUE CLUSTERED INDEX orders_c1 ON
orders(o_w_id, o_d_id, o_id)
ON misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',

```



```

        DATEDIFF(second, @startdate, @enddate)
GO

-----
--                                     --
-- File:  IDXSTKCL.SQL                 --
-- Microsoft TPC-C Benchmark Kit Ver. 4.62
--
-- Copyright Microsoft, 2005          --
--
-- Creates clustered index on stock table
--
-----

USE tpcc
GO

DECLARE @startdate    DATETIME,
        @enddate      DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE
            name = 'stock_c1' )
    DROP INDEX stock.stock_c1

CREATE UNIQUE CLUSTERED INDEX stock_c1 ON
stock(s_i_id, s_w_id)
    ON cs_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

-----
--                                     --
-- File:  IDXWARCL.SQL                 --
-- Microsoft TPC-C Benchmark Kit Ver. 4.62
--
-- Copyright Microsoft, 2005          --
--
-- Creates clustered index on warehouse table
--
-----

USE tpcc
GO

DECLARE @startdate    DATETIME,
        @enddate      DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE
            name = 'warehouse_c1' )
    DROP INDEX warehouse.warehouse_c1

CREATE UNIQUE CLUSTERED INDEX warehouse_c1
ON warehouse(w_id)
    WITH FILLFACTOR=100 ON misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

### Non Clustered Index Creation Scripts

```

-----
--                                     --
-- File:  IDXCUSNC.SQL                 --
-- Microsoft TPC-C Benchmark Kit Ver. 4.62
--
-- Copyright Microsoft, 2005          --
--
-- Creates non-clustered index on customer
table
--
-----

USE tpcc
GO

DECLARE @startdate    DATETIME,
        @enddate      DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE
            name = 'customer_nc1' )
    DROP INDEX customer.customer_nc1

CREATE UNIQUE NONCLUSTERED INDEX
customer_nc1 ON customer(c_w_id, c_d_id, c_last,
                        c_first, c_id)
    ON cs_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

-----
--                                     --
-- File:  IDXORDNC.SQL                 --
-- Microsoft TPC-C Benchmark Kit Ver. 4.62
--
-- Copyright Microsoft, 2005          --
--
-- Creates non-clustered index on orders table
--
-----

USE tpcc
GO

DECLARE @startdate    DATETIME,
        @enddate      DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE
            name = 'orders_nc1' )
    DROP INDEX orders.orders_nc1

CREATE INDEX orders_nc1 ON orders(o_w_id, o_d_id,
                                o_c_id, o_id)
    ON misc_fg

```

```

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

## B.3 Stored Procedures

### Create\_NewOrd.sql

```

-----
--                                     --
-- File:  NEWORD.SQL                   --
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005          --
--
-- Creates neworder stored procedure
--
-- Interface Level:  4.20.000         --
--
-----

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
            name = 'tpcc_neworder' )
    DROP PROCEDURE tpcc_neworder
GO

CREATE PROCEDURE tpcc_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      smallmoney,
        @d_tax      smallmoney,
        @c_last     char(16),
        @c_credit   char(2),
        @c_discount smallmoney,
        @i_price    smallmoney,
        @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_o_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 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 (repeatableread)
WHERE  i_id    = @li_id

```

```

-----
-- update stock values
-----

```

```

UPDATE stock
SET  s_ytd      = s_ytd + @li_qty,
     @s_quantity = s_quantity - @li_qty +

```

```

CASE WHEN (s_quantity -
@li_qty < 10) THEN 91 ELSE 0 END,
     s_order_cnt = s_order_cnt + 1,
     s_remote_cnt = s_remote_cnt +

```

```

CASE WHEN (@li_s_w_id =
@w_id) THEN 0 ELSE 1 END,
     @s_data     = s_data,
     @s_dist     = CASE @d_id

```

```

    WHEN 1 THEN s_dist_01
    WHEN 2 THEN s_dist_02
    WHEN 3 THEN s_dist_03
    WHEN 4 THEN s_dist_04
    WHEN 5 THEN s_dist_05
    WHEN 6 THEN s_dist_06
    WHEN 7 THEN s_dist_07

```

```

    WHEN 8 THEN s_dist_08
    WHEN 9 THEN s_dist_09
    WHEN 10 THEN s_dist_10
END
WHERE  s_i_id   = @li_id AND
       s_w_id   = @li_s_w_id

```

```

-----
-- if there actually is a stock (and item) with these ids,
go to work
-----

```

```

IF (@@rowcount > 0)
BEGIN

```

```

-- insert order_line data (using data from item and
stock)

```

```

INSERT INTO order_line VALUES( @o_id,
                                @d_id,
                                @w_id,
                                @li_no,
                                @li_id,
                                'dec 31, 1899',
                                @i_price * @li_qty,
                                @li_s_w_id,
                                @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_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,
        0,
        @o_ol_cnt,
        @o_all_local,
        @o_entry_d)

-----
-- 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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

```

### TPcc\_neworder\_new.sql

```

-----
-- File: TPCC_NEWORDER_NEW.SQL
--
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005
--
-- This acid stored procedure implements the
neworder --
-- transaction. It outputs timestamps at the
--
-- beginning of the transaction, before the
commit --
-- delay, and after the commit.
--
-----

```

```

SET QUOTED_IDENTIFIER OFF
SET ANSI_NULLS OFF
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'tpcc_neworder_new' )
DROP PROCEDURE tpcc_neworder_new
GO

-- neworder_new v2.5 6/23/05 PeterCa
-- 1q stock/order_line/client. upd district & ins
neworder.
-- cust/warehouse select together, ins order separate
-- uses rownumber to distinct w any transform
-- uses in-memory sort for distinct on iid,wid
-- uses charindex
-- will rollback if (@i_idX,@s_w_idX pairs not unique)
OR (@i_idX not unique).

CREATE PROCEDURE tpcc_neworder_new
    @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
BEGIN
DECLARE @o_id int,
        @d_tax smallmoney,
        @o_entry_d datetime,
        @commit_flag tinyint

BEGIN TRANSACTION n
-- get district tax and next available order id and
update
-- insert corresponding row into new-order table
-- plus initialize local variables

UPDATE district

```

```

SET @d_tax = d_tax,
    d_next_o_id = d_next_o_id + 1,

    @o_entry_d = GETDATE(),
    @commit_flag = 1
OUTPUT deleted.d_next_o_id,
       @d_id,
       @w_id
INTO new_order
WHERE d_w_id = @w_id AND
      d_id = @d_id

-- update stock from stock join (item join (params))
-- output to orderline, output to client
-- NOTE: @@rowcount != @ol_o_cnt
-- if (@i_idX,@s_w_idX pairs not unique)
OR (@i_idX not unique).

UPDATE stock
SET s_ytd = s_ytd + info.ol_qty,
    s_quantity = s_quantity - info.ol_qty +
CASE WHEN (s_quantity -
info.ol_qty < 10) THEN 91 ELSE 0 END,
    s_order_cnt = s_order_cnt + 1,
    s_remote_cnt = s_remote_cnt +
CASE WHEN (info.w_id = @w_id) THEN 0 ELSE 1 END

OUTPUT @o_id,
       @d_id,
       @w_id,
       info.lino,
       info.i_id,
       "dec 31, 1899",
       info.i_price * info.ol_qty,
       info.w_id,
       info.ol_qty,
CASE @d_id WHEN 1 THEN
inserted.s_dist_01
      WHEN 2 THEN inserted.s_dist_02
      WHEN 3 THEN inserted.s_dist_03
      WHEN 4 THEN inserted.s_dist_04
      WHEN 5 THEN inserted.s_dist_05
      WHEN 6 THEN inserted.s_dist_06
      WHEN 7 THEN inserted.s_dist_07
      WHEN 8 THEN inserted.s_dist_08
      WHEN 9 THEN inserted.s_dist_09
      WHEN 10 THEN inserted.s_dist_10

END
INTO order_line

OUTPUT info.i_name,inserted.s_quantity,
CASE WHEN
((charindex("ORIGINAL",info.i_data) > 0) AND
(charindex("ORIGINAL",inserted.s_data) > 0) )
THEN "B" ELSE "G" END,
info.i_price,
info.i_price*info.ol_qty
FROM stock INNER JOIN
(SELECT iid,
      wid,
      lino,
      ol_qty,
      i_price,
      i_name,
      i_data
FROM (SELECT iid,
      wid,
      lino,
      qty,
row_number() OVER (PARTITION BY iid,wid ORDER
BY iid,wid)

```

```

FROM (SELECT
@i_id1,@s_w_id1,1,@ol_qty1 UNION ALL
SELECT
@i_id2,@s_w_id2,2,@ol_qty2 UNION ALL
SELECT
@i_id3,@s_w_id3,3,@ol_qty3 UNION ALL
SELECT
@i_id4,@s_w_id4,4,@ol_qty4 UNION ALL
SELECT
@i_id5,@s_w_id5,5,@ol_qty5 UNION ALL
SELECT
@i_id6,@s_w_id6,6,@ol_qty6 UNION ALL
SELECT
@i_id7,@s_w_id7,7,@ol_qty7 UNION ALL
SELECT
@i_id8,@s_w_id8,8,@ol_qty8 UNION ALL
SELECT
@i_id9,@s_w_id9,9,@ol_qty9 UNION ALL
SELECT
@i_id10,@s_w_id10,10,@ol_qty10 UNION ALL
SELECT
@i_id11,@s_w_id11,11,@ol_qty11 UNION ALL
SELECT
@i_id12,@s_w_id12,12,@ol_qty12 UNION ALL
SELECT
@i_id13,@s_w_id13,13,@ol_qty13 UNION ALL
SELECT
@i_id14,@s_w_id14,14,@ol_qty14 UNION ALL
SELECT
@i_id15,@s_w_id15,15,@ol_qty15) AS
uol(iid,wid,lino,qty)
) AS
o1(iid,wid,lino,ol_qty,rownum)
INNER JOIN
item (repeatableread) ON i_id =
iid AND -- filters out invalid items
rownum = 1
) AS
info(i_id,w_id,lino,ol_qty,i_price,i_name,i_data)
ON s_i_id = info.i_id AND
s_w_id = info.w_id

IF (@@rowcount <> @o_ol_cnt) -- must have an
invalid item
SELECT @commit_flag = 0 -- 2.4.2.3 requires
rest to proceed

-- insert fresh row into orders table
INSERT INTO orders VALUES ( @o_id,
@o_id,
@w_id,
@o_id,
0,
@o_ol_cnt,
@o_all_local,
@o_entry_d)

-- get customer last name, discount, and credit
rating
-- get warehouse tax
-- return order_data to client

SELECT w_tax,
@d_tax,
@o_id,
c_last,
c_discount,
c_credit,
@o_entry_d,
@commit_flag
FROM warehouse(repeatableread),
customer(repeatableread)
WHERE w_id = @w_id AND
c_id = @c_id AND
c_w_id = @w_id AND

```

```

c_d_id = @d_id
-- @@rowcount checks that previous
select found a valid customer
IF (@@commit_flag = 1) AND (@@rowcount = 1)
COMMIT TRANSACTION n
ELSE -- all that work for nothing.
ROLLBACK TRANSACTION n

END
GO

OrdStat.sql
-----
-- File: ORDSTAT.SQL --
-- Microsoft TPC-C Benchmark Kit Ver. 4.63 --
-- Copyright Microsoft, 2005 --
-- Creates order status stored procedure --
-- Interface Level: 4.20.000 --
-----

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'tpcc_orderstatus' )
DROP PROCEDURE tpcc_orderstatus
GO

CREATE PROCEDURE tpcc_orderstatus
@w_id int,
@d_id tinyint,
@c_id int,
@c_last char(16) = ""
AS
DECLARE @c_balance money,
@c_first char(16),
@c_middle char(2),
@o_id int,
@o_entry_d datetime,
@o_carrier_id smallint,
@cnt smallint

BEGIN TRANSACTION 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
@w_id = @w_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 TRANSACTION 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

```

### delivery.sql

```

-----
-- File: DELIVERY.SQL --
-- Microsoft TPC-C Benchmark Kit Ver. 4.63 --
-- Copyright Microsoft, 2005 --
-- Creates delivery stored procedure --
-- Interface Level: 4.20.000 --
-----
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'tpcc_delivery' )
DROP PROCEDURE tpcc_delivery
GO

CREATE PROC tpcc_delivery
    @w_id int,
    @o_carrier_id smallint
AS

DECLARE @d_id tinyint,
        @o_id int,
        @c_id int,
        @total money,
        @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 TRANSACTION d
WHILE (@d_id < 10)
BEGIN

```

```

SELECT @oid1 = @d_id + 1,
       @o_id = 0

SELECT TOP 1
    @o_id = no_o_id
FROM new_order WITH (serializable uplock)
WHERE no_w_id = @w_id AND
      no_d_id = @d_id
ORDER BY no_o_id ASC

IF (@@rowcount <> 0)
BEGIN
-- claim the order for this district
DELETE new_order
WHERE no_w_id = @w_id AND
      no_d_id = @d_id AND
      no_o_id = @o_id

-- set carrier_id on this order (and get
customer id)
UPDATE orders
SET o_carrier_id = @o_carrier_id,
    @c_id = o_c_id
WHERE o_w_id = @w_id AND
      o_d_id = @d_id AND
      o_id = @o_id

-- set date in all lineitems for this order (and
sum amounts)
UPDATE order_line
SET ol_delivery_d = GETDATE(),
    @total = @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 TRANSACTION d

-- return delivery data to client

SELECT @oid1,
       @oid2,

```

```

@oid3,
@oid5,
@oid6,
@oid7,

@oid8,
@oid9,
@oid10
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

```

### createpaymentproc.sql

```

-----
-- File: PAYMENT.SQL --
-- Microsoft TPC-C Benchmark Kit Ver. 4.63 --
-- Copyright Microsoft, 2005 --
-- Creates payment stored procedure --
-- Interface Level: 4.20.000 --
-----
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'tpcc_payment' )
DROP PROCEDURE tpcc_payment
GO

CREATE PROCEDURE tpcc_payment
    @w_id int,
    @c_w_id int,
    @h_amount smallmoney,
    @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 money,
@c_balance money,
@c_discount smallmoney,
@c_data char(42),
@datetime datetime,
@w_ytd money,
@d_ytd money,
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int

SELECT @screen_data = ""

BEGIN TRANSACTION 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 AND
@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 AND
@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_lim = c_credit_lim,
@c_discount = c_discount,
@c_since = c_since,
@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)

-- update customer info
UPDATE customer
SET c_data = @c_data +
substring(c_data, 1, 458),
@screen_data = @c_data +
substring(c_data, 1, 158)
WHERE c_id = @c_id AND
c_w_id = @c_w_id AND
c_d_id = @c_d_id
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,
@datetime,
@h_amount,
@w_name + ' ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT @c_id,
@c_last,
@datetime,
@w_street_1,
@w_street_2,
@w_city,

```

```

@w_state,
@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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

-----
-- File: STOCKLEV.SQL --
-- Microsoft TPC-C Benchmark Kit Ver. 4.63 --
-- Copyright Microsoft, 2005 --
-- Creates stock level stored procedure --
-- Interface Level: 4.20.000 --
-----

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'tpcc_stocklevel' )
DROP PROCEDURE tpcc_stocklevel
GO

CREATE PROCEDURE tpcc_stocklevel
@w_id int,
@d_id tinyint,
@threshold smallint

AS
DECLARE @o_id_low int,
@o_id_high int

```

#### createstocklevproc.sql

```
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
OPTION(OPTION ORDER GROUP)
GO
```

```
SET QUOTED_IDENTIFIER OFF
GO
```

```
SET ANSI_NULLS ON
GO
```





[DMA]

| Resource | Device | Status |
|----------|--------|--------|
|----------|--------|--------|

[Forced Hardware]

| Device | PNP Device ID |
|--------|---------------|
|--------|---------------|

[I/O]

| Resource              | Device  | Status |
|-----------------------|---------|--------|
| 0x00000000-0x00001FFF | PCI bus | OK     |

|                       |         |    |
|-----------------------|---------|----|
| 0x00000000-0x00001FFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x00000000-0x00001FFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x00000000-0x00001FFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x00000000-0x00001FFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x00000000-0x00001FFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x00000000-0x00001FFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x00000000-0x00001FFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x00000000-0x00001FFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |                    |    |
|-----------------------|--------------------|----|
| 0x00001000-0x000010FF | RADEON 7000 SERIES | OK |
|-----------------------|--------------------|----|

|                       |                    |    |
|-----------------------|--------------------|----|
| 0x000003B0-0x000003BB | RADEON 7000 SERIES | OK |
|-----------------------|--------------------|----|

|                       |                    |    |
|-----------------------|--------------------|----|
| 0x000003C0-0x000003DF | RADEON 7000 SERIES | OK |
|-----------------------|--------------------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x00002000-0x00003FFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |  |    |
|-----------------------|--|----|
| 0x00002000-0x00003FFF | Emulex LightPulse LP1050, Storport Miniport Driver | OK |
|-----------------------|--|----|

|                       |  |    |
|-----------------------|--|----|
| 0x00002100-0x000021FF | Emulex LightPulse LP1050, Storport Miniport Driver | OK |
|-----------------------|--|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x00004000-0x00005FFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0x00004000-0x00005FFF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0x00004100-0x000041FF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x00006000-0x00007FFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0x00006000-0x00007FFF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0x00006100-0x000061FF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x00008000-0x00009FFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |                             |    |
|-----------------------|-----------------------------|----|
| 0x00008000-0x00009FFF | Smart Array P600 Controller | OK |
|-----------------------|-----------------------------|----|

|                       |  |    |
|-----------------------|--|----|
| 0x00008140-0x0000817F | Intel(R) PRO/1000 MT Dual Port Server Adapter #3 | OK |
|-----------------------|--|----|

|                       |  |    |
|-----------------------|--|----|
| 0x00008100-0x0000813F | Intel(R) PRO/1000 MT Dual Port Server Adapter #4 | OK |
|-----------------------|--|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x0000A000-0x0000BFFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0x0000A000-0x0000BFFF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0x0000A100-0x0000A1FF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x0000C000-0x0000DFFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0x0000C000-0x0000DFFF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0x0000C100-0x0000C1FF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x0000E000-0x0000FFFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0x0000E000-0x0000FFFF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0x0000E100-0x0000E1FF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

[IRQs]

| Resource | Device                          | Status |
|----------|---------------------------------|--------|
| IRQ 25   | Microsoft ACPI-Compliant System | OK     |

|        |                                     |    |
|--------|-------------------------------------|----|
| IRQ 17 | NEC PCI to USB Open Host Controller | OK |
|--------|-------------------------------------|----|

|        |                                     |    |
|--------|-------------------------------------|----|
| IRQ 18 | NEC PCI to USB Open Host Controller | OK |
|--------|-------------------------------------|----|

|        |  |    |
|--------|--|----|
| IRQ 19 | Standard Enhanced PCI to USB Host Controller | OK |
|--------|--|----|

|        |                    |    |
|--------|--------------------|----|
| IRQ 20 | RADEON 7000 SERIES | OK |
|--------|--------------------|----|

|        |                            |    |
|--------|----------------------------|----|
| IRQ 24 | Communications Port (COM1) | OK |
|--------|----------------------------|----|

|        |  |    |
|--------|--|----|
| IRQ 27 | Emulex LightPulse LP1050, Storport Miniport Driver | OK |
|--------|--|----|

|        |  |    |
|--------|--|----|
| IRQ 28 | Emulex LightPulse LP1050, Storport Miniport Driver | OK |
|--------|--|----|

|        |                              |    |
|--------|------------------------------|----|
| IRQ 38 | QLogic Fibre Channel Adapter | OK |
|--------|------------------------------|----|

|        |                              |    |
|--------|------------------------------|----|
| IRQ 39 | QLogic Fibre Channel Adapter | OK |
|--------|------------------------------|----|

|        |                              |    |
|--------|------------------------------|----|
| IRQ 49 | QLogic Fibre Channel Adapter | OK |
|--------|------------------------------|----|

|        |                              |    |
|--------|------------------------------|----|
| IRQ 50 | QLogic Fibre Channel Adapter | OK |
|--------|------------------------------|----|

|        |                             |    |
|--------|-----------------------------|----|
| IRQ 60 | Smart Array P600 Controller | OK |
|--------|-----------------------------|----|

|        |  |    |
|--------|--|----|
| IRQ 64 | Intel(R) PRO/1000 MT Dual Port Server Adapter #3 | OK |
|--------|--|----|

|        |  |    |
|--------|--|----|
| IRQ 65 | Intel(R) PRO/1000 MT Dual Port Server Adapter #4 | OK |
|--------|--|----|

|        |                              |    |
|--------|------------------------------|----|
| IRQ 71 | QLogic Fibre Channel Adapter | OK |
|--------|------------------------------|----|

|        |                              |    |
|--------|------------------------------|----|
| IRQ 72 | QLogic Fibre Channel Adapter | OK |
|--------|------------------------------|----|

|        |                              |    |
|--------|------------------------------|----|
| IRQ 82 | QLogic Fibre Channel Adapter | OK |
|--------|------------------------------|----|

|        |                              |    |
|--------|------------------------------|----|
| IRQ 83 | QLogic Fibre Channel Adapter | OK |
|--------|------------------------------|----|

|        |                              |    |
|--------|------------------------------|----|
| IRQ 93 | QLogic Fibre Channel Adapter | OK |
|--------|------------------------------|----|

|        |                              |    |
|--------|------------------------------|----|
| IRQ 94 | QLogic Fibre Channel Adapter | OK |
|--------|------------------------------|----|

[Memory]

| Resource              | Device  | Status |
|-----------------------|---|--------|
| 0xFF5BC000-0xFF5BC003 | HP Baseboard Management Controller Interface Device | OK     |

|                 |         |    |
|-----------------|---------|----|
| 0xA0000-0xFFFFF | PCI bus | OK |
|-----------------|---------|----|

|                 |                    |    |
|-----------------|--------------------|----|
| 0xA0000-0xFFFFF | RADEON 7000 SERIES | OK |
|-----------------|--------------------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x80000000-0x8FFFFFFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |                    |    |
|-----------------------|--------------------|----|
| 0x80000000-0x8FFFFFFF | RADEON 7000 SERIES | OK |
|-----------------------|--------------------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0xFF5E2000-0xFF5E2007 | PCI bus | OK |
|-----------------------|---------|----|

|                       |                     |    |
|-----------------------|---------------------|----|
| 0xFF5E2000-0xFF5E2007 | Communications Port | OK |
|-----------------------|---------------------|----|

|                       |                                    |    |
|-----------------------|------------------------------------|----|
| 0x84054000-0x84054FFF | HP Management Shared Memory Device | OK |
|-----------------------|------------------------------------|----|

|                       |                                    |    |
|-----------------------|------------------------------------|----|
| 0x84020000-0x8403FFFF | HP Management Shared Memory Device | OK |
|-----------------------|------------------------------------|----|

|                       |                         |    |
|-----------------------|-------------------------|----|
| 0x84053000-0x84053FFF | HP Management Processor | OK |
|-----------------------|-------------------------|----|

|                       |                                     |    |
|-----------------------|-------------------------------------|----|
| 0x84052000-0x84052FFF | NEC PCI to USB Open Host Controller | OK |
|-----------------------|-------------------------------------|----|

|                       |                                     |    |
|-----------------------|-------------------------------------|----|
| 0x84051000-0x84051FFF | NEC PCI to USB Open Host Controller | OK |
|-----------------------|-------------------------------------|----|

|                       |  |    |
|-----------------------|--|----|
| 0x84050000-0x840500FF | Standard Enhanced PCI to USB Host Controller | OK |
|-----------------------|--|----|

|                       |                    |    |
|-----------------------|--------------------|----|
| 0x84040000-0x8404FFFF | RADEON 7000 SERIES | OK |
|-----------------------|--------------------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0x90000000-0x9FFFFFFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |  |    |
|-----------------------|--|----|
| 0x90000000-0x9FFFFFFF | Emulex LightPulse LP1050, Storport Miniport Driver | OK |
|-----------------------|--|----|

|                       |  |    |
|-----------------------|--|----|
| 0x90003000-0x90003FFF | Emulex LightPulse LP1050, Storport Miniport Driver | OK |
|-----------------------|--|----|

|                       |  |    |
|-----------------------|--|----|
| 0x90002000-0x900020FF | Emulex LightPulse LP1050, Storport Miniport Driver | OK |
|-----------------------|--|----|

|                       |  |    |
|-----------------------|--|----|
| 0x90001000-0x90001FFF | Emulex LightPulse LP1050, Storport Miniport Driver | OK |
|-----------------------|--|----|

|                       |         |    |
|-----------------------|---------|----|
| 0xA0000000-0xAFFFFFFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0xA0041000-0xA0041FFF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0xA0040000-0xA0040FFF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0xB0000000-0xBFFFFFFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0xB0041000-0xB0041FFF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0xB0040000-0xB0040FFF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0xC0000000-0xCFFFFFFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |                             |    |
|-----------------------|-----------------------------|----|
| 0xC01C0000-0xC01C1FFF | Smart Array P600 Controller | OK |
|-----------------------|-----------------------------|----|

|                       |                             |    |
|-----------------------|-----------------------------|----|
| 0xC0140000-0xC017FFFF | Smart Array P600 Controller | OK |
|-----------------------|-----------------------------|----|

|                       |  |    |
|-----------------------|--|----|
| 0xC01A0000-0xC01BFFFF | Intel(R) PRO/1000 MT Dual Port Server Adapter #3 | OK |
|-----------------------|--|----|

|                       |  |    |
|-----------------------|--|----|
| 0xC00C0000-0xC00FFFFF | Intel(R) PRO/1000 MT Dual Port Server Adapter #3 | OK |
|-----------------------|--|----|

|                       |  |    |
|-----------------------|--|----|
| 0xC0180000-0xC019FFFF | Intel(R) PRO/1000 MT Dual Port Server Adapter #4 | OK |
|-----------------------|--|----|

|                       |  |    |
|-----------------------|--|----|
| 0xC0040000-0xC007FFFF | Intel(R) PRO/1000 MT Dual Port Server Adapter #4 | OK |
|-----------------------|--|----|

|                       |         |    |
|-----------------------|---------|----|
| 0xD0000000-0xDFFFFFFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0xD0041000-0xD0041FFF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0xD0040000-0xD0040FFF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0xE0000000-0xEFFFFFFF | PCI bus | OK |
|-----------------------|---------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0xE0041000-0xE0041FFF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |                              |    |
|-----------------------|------------------------------|----|
| 0xE0040000-0xE0040FFF | QLogic Fibre Channel Adapter | OK |
|-----------------------|------------------------------|----|

|                       |         |    |
|-----------------------|---------|----|
| 0xF0000000-0xFFFFFFFF | PCI bus | OK |
|-----------------------|---------|----|

0xF0041000-0xF0041FFF QLogic Fibre Channel Adapter OK  
 0xF0040000-0xF0040FFF QLogic Fibre Channel Adapter OK

[Components]

[Multimedia]

[Audio Codecs]

| CODEC                            | Manufacturer                   | Description                        | Status                  |
|----------------------------------|--------------------------------|------------------------------------|-------------------------|
|                                  | File                           | Version                            | Size                    |
|                                  | Creation Date                  |                                    |                         |
| c:\windows\system32\msg711.acm   | Microsoft Corporation          | OK                                 |                         |
| M                                | C:\windows\system32\MSG711.AC  | 5.2.3790.0 (srv03_rtm.030324-2048) | 33.00 KB (33,792 bytes) |
|                                  |                                | 3/25/2005 4:00 AM                  |                         |
| c:\windows\system32\msadp32.acm  | Microsoft Corporation          | OK                                 |                         |
| M                                | C:\windows\system32\MSADP32.AC | 5.2.3790.0 (srv03_rtm.030324-2048) | 49.00 KB (50,176 bytes) |
|                                  |                                | 3/25/2005 4:00 AM                  |                         |
| c:\windows\system32\imaadp32.acm | Microsoft Corporation          | OK                                 |                         |
| CM                               | C:\windows\system32\IMAADP32.A | 5.2.3790.0 (srv03_rtm.030324-2048) | 55.00 KB (56,320 bytes) |
|                                  |                                | 3/25/2005 4:00 AM                  |                         |
| c:\windows\system32\msgsm32.acm  | Microsoft Corporation          | OK                                 |                         |
| CM                               | C:\windows\system32\MSGSM32.A  | 5.2.3790.0 (srv03_rtm.030324-2048) | 66.50 KB (68,096 bytes) |
|                                  |                                | 3/25/2005 4:00 AM                  |                         |
| c:\windows\system32\tssoft32.acm | DSP GROUP, INC.                | OK                                 |                         |
| CM                               | C:\windows\system32\TSSOFT32.A | 1.01                               | 29.00 KB (29,696 bytes) |
|                                  |                                | 3/25/2005 4:00 AM                  |                         |

[Video Codecs]

| CODEC                            | Manufacturer                   | Description                        | Status                  |
|----------------------------------|--------------------------------|------------------------------------|-------------------------|
|                                  | File                           | Version                            | Size                    |
|                                  | Creation Date                  |                                    |                         |
| c:\windows\system32\msrle32.dll  | Microsoft Corporation          | OK                                 |                         |
| L                                | C:\windows\system32\MSRLE32.DL | 5.2.3790.0 (srv03_rtm.030324-2048) | 24.50 KB (25,088 bytes) |
|                                  |                                | 3/25/2005 4:00 AM                  |                         |
| c:\windows\system32\msvidc32.dll | Microsoft Corporation          | OK                                 |                         |
| LL                               | C:\windows\system32\MSVIDC32.D | 5.2.3790.0 (srv03_rtm.030324-2048) | 67.00 KB (68,608 bytes) |
|                                  |                                | 3/25/2005 4:00 AM                  |                         |

[CD-ROM]

| Item         | Value                    |
|--------------|--------------------------|
| Drive        | D:                       |
| Description  | CD-ROM Drive             |
| Media Loaded | Yes                      |
| Media Type   | CD-ROM                   |
| Name         | TEAC DV-28E-N USB Device |

Manufacturer (Standard CD-ROM drives)  
 Status OK  
 Transfer Rate 3247.06 kbytes/sec  
 SCSI Target ID Not Available

PNP Device ID  
 USBSTOR\CDROM&VEN\_TEAC&PRO  
 D\_DV-28E-N&REV\_C.6B\A6002000001&0

Driver  
 c:\windows\system32\drivers\cdrom.sys (5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447), 144.50 KB (147,968 bytes), 3/25/2005 4:00 AM)

[Sound Device]

| Item | Value |
|------|-------|
|------|-------|

[Display]

| Item                | Value   |
|---------------------|---|
| Name                | RADEON 7000 SERIES  |
| PNP Device ID       | PCI\VEN_1002&DEV_5159&SUBSYS_1292103C&REV_00\4&4F5EBC7&0&20 |
| Adapter Type        | Not Available   |
| Adapter Description | ATI Technologies Inc.                                       |
| Adapter RAM         | Not Available   |
| Installed Drivers   | Not Available   |
| Driver Version      | Not Available   |
| INF File            | 5.2.3763.0 (oem3.inf section)                               |

Color Planes ati2mtag\_RV100  
 Color Table Entries Not Available  
 Resolution Not Available  
 Bits/Pixel Not Available  
 Memory Address 0x80000000-0x8FFFFFFF  
 I/O Port 0x00001000-0x000010FF  
 Memory Address 0x84040000-0x8404FFFF  
 IRQ Channel IRQ 20  
 I/O Port 0x000003B0-0x000003BB  
 I/O Port 0x000003C0-0x000003DF  
 Memory Address 0xA0000-0xFFFFF

Driver  
 c:\windows\system32\drivers\ati2mtag.sys (6.14.10.6368, 1.46 MB (1,534,976 bytes), 6/20/2003 10:18 AM)

[Infrared]

| Item | Value |
|------|-------|
|------|-------|

[Input]

[Keyboard]

| Item        | Value                      |
|-------------|----------------------------|
| Description | USB Human Interface Device |

| Name                    | Enhanced (101- or 102-key)  |
|-------------------------|---|
| Layout                  | 00000409  |
| PNP Device ID           | USB\VID_03F0&PID_1126&MI_00\7&23D9570F&0&0000   |
| Number of Function Keys | 12  |
| Driver                  | c:\windows\system32\drivers\hidusb.sys (5.2.3790.0 (srv03_rtm.030324-2048), 32.00 KB (32,768 bytes), 3/25/2005 4:00 AM) |

[Pointing Device]

| Item                       | Value  |
|----------------------------|--|
| Hardware Type              | USB Human Interface Device                     |
| Number of Buttons          | 3  |
| Status                     | OK   |
| PNP Device ID              | USB\VID_03F0&PID_1126&MI_01\7&23D9570F&0&00001 |
| Power Management Supported | No   |

| Item                   | Value   |
|------------------------|---|
| Double Click Threshold | 6   |
| Handedness             | Right Handed Operation  |
| Driver                 | c:\windows\system32\drivers\hidusb.sys (5.2.3790.0 (srv03_rtm.030324-2048), 32.00 KB (32,768 bytes), 3/25/2005 4:00 AM) |

[Modem]

| Item | Value |
|------|-------|
|------|-------|

[Network]

[Adapter]

| Item                | Value  |
|---------------------|--|
| Name                | [00000001] Intel(R) PRO/1000 MT Dual Port Server Adapter |
| Adapter Type        | Not Available  |
| Product Type        | Intel(R) PRO/1000 MT Dual Port Server Adapter            |
| Installed           | Yes  |
| PNP Device ID       | Not Available  |
| Last Reset          | 7/7/2006 4:19 PM   |
| Index               | 1  |
| Service Name        | E1000  |
| 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       | Value             |
|------------|-------------------|
| [00000002] | RAS Async Adapter |

|               |                   |
|---------------|-------------------|
| Adapter Type  | Not Available     |
| Product Type  | RAS Async Adapter |
| Installed     | Yes               |
| PNP Device ID | Not Available     |
| Last Reset    | 7/7/2006 4:19 PM  |
| 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 TypeNot Available  
Product TypeWAN Miniport (L2TP)  
Installed Yes  
PNP Device ID  
ROOT\MS\_L2TPMINIIMPORT\0000

Last Reset 7/7/2006 4:19 PM  
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 (5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447), 188.00 KB (192,512 bytes), 3/25/2005 4:00 AM)

Name [00000004] WAN Miniport (PPTP)

Adapter TypeWide Area Network (WAN)  
Product TypeWAN Miniport (PPTP)  
Installed Yes  
PNP Device ID  
ROOT\MS\_PPTPMINIIMPORT\0000

Last Reset 7/7/2006 4:19 PM  
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\rasppp.sys (5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447), 176.00 KB (180,224 bytes), 3/25/2005 4:00 AM)

Name [00000005] WAN Miniport (PPPOE)

Adapter TypeWide Area Network (WAN)  
Product TypeWAN Miniport (PPPOE)  
Installed Yes  
PNP Device ID  
ROOT\MS\_PPPOEMINIIMPORT\0000

Last Reset 7/7/2006 4:19 PM  
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

MAC Address 33:50:6F:45:30:30

Driver  
c:\windows\system32\drivers\rasppoe.sys (5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447), 131.00 KB (134,144 bytes), 3/25/2005 4:00 AM)

Name [00000006] Direct Parallel

Adapter TypeNot Available  
Product TypeDirect Parallel  
Installed Yes  
PNP Device ID  
ROOT\MS\_PTIMINIIMPORT\0000

Last Reset 7/7/2006 4:19 PM  
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 (5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447), 54.50 KB (55,808 bytes), 3/25/2005 4:00 AM)

Name [00000007] WAN Miniport (IP)

Adapter TypeNot Available  
Product TypeWAN Miniport (IP)  
Installed Yes  
PNP Device ID  
ROOT\MS\_NDISWANIP\0000

Last Reset 7/7/2006 4:19 PM  
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 (5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447), 266.50 KB (272,896 bytes), 3/25/2005 4:00 AM)

Name [00000008] Intel(R) PRO/1000 MT

Dual Port Server Adapter  
Adapter TypeNot Available  
Product TypeIntel(R) PRO/1000 MT Dual Port Server Adapter  
Installed Yes  
PNP Device ID Not Available

Last Reset 7/7/2006 4:19 PM  
Index 8  
Service Name E1000

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

MAC Address Not Available Not Available

Name [00000009] Intel(R) PRO/1000 MT  
Dual Port Server Adapter  
Adapter TypeEthernet 802.3  
Product TypeIntel(R) PRO/1000 MT Dual Port Server Adapter  
Installed Yes  
PNP Device ID

PCI\VEN\_8086&DEV\_1079&SUBSYS\_12A6103C&REV\_03\4&15291AB&0&10  
Last Reset 7/7/2006 4:19 PM  
Index 9

Service Name E1000  
IP Address 15.1.104.2  
IP Subnet 255.255.0.0  
Default IP Gateway Not Available  
DHCP Enabled No  
DHCP Server Not Available  
DHCP Lease Expires Not Available  
DHCP Lease Obtained Not Available

MAC Address 00:30:6E:5D:AC:4C  
Memory Address 0xC01A0000-0xC01BFFFF  
Memory Address 0xC00C0000-0xC00FFFFF  
I/O Port 0x00008140-0x0000817F  
IRQ Channel IRQ 64  
Driver

c:\windows\system32\drivers\e1000645.sys (8.5.14.0 built by: WinDDK, 525.50 KB (538,112 bytes), 3/8/2005 9:37 AM)

Name [00000010] Intel(R) PRO/1000 MT

Dual Port Server Adapter  
Adapter TypeEthernet 802.3  
Product TypeIntel(R) PRO/1000 MT Dual Port Server Adapter  
Installed Yes  
PNP Device ID

PCI\VEN\_8086&DEV\_1079&SUBSYS\_12A6103C&REV\_03\4&15291AB&0&11  
Last Reset 7/7/2006 4:19 PM  
Index 10  
Service Name E1000

IP Address 15.1.104.1  
IP Subnet 255.255.0.0  
Default IP Gateway Not Available  
DHCP Enabled No  
DHCP Server Not Available  
DHCP Lease Expires Not Available  
DHCP Lease Obtained Not Available

MAC Address 00:30:6E:5D:AC:4D  
Memory Address 0xC0180000-0xC019FFFF  
Memory Address 0xC0040000-0xC007FFFF  
I/O Port 0x00008100-0x0000813F  
IRQ Channel IRQ 65  
Driver

c:\windows\system32\drivers\e1000645.sys (8.5.14.0 built by: WinDDK, 525.50 KB (538,112 bytes), 3/8/2005 9:37 AM)

[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 MSAFD Tcip [UDP/IP]  
 Connectionless Service Yes  
 Guarantees Delivery No  
 Guarantees Sequencing No  
 Maximum Address Size 16 bytes  
 Maximum Message Size 63.93 KB (65,467 bytes)  
 Message Oriented Yes  
 Minimum Address Size 16 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting Yes

Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption No  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting Yes

Name RSVP UDP Service Provider  
 Connectionless Service Yes  
 Guarantees Delivery No  
 Guarantees Sequencing No  
 Maximum Address Size 16 bytes  
 Maximum Message Size 63.93 KB (65,467 bytes)  
 Message Oriented Yes  
 Minimum Address Size 16 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting Yes  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption Yes  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting Yes

Name RSVP TCP Service Provider  
 Connectionless Service No  
 Guarantees Delivery Yes  
 Guarantees Sequencing Yes  
 Maximum Address Size 16 bytes  
 Maximum Message Size 0 bytes  
 Message Oriented No  
 Minimum Address Size 16 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting No  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption Yes  
 Supports Expedited Data Yes  
 Supports Graceful Closing Yes  
 Supports Guaranteed Bandwidth No

Supports Multicasting No

[WinSock]

Item Value  
 File c:\windows\system32\wsock32.dll  
 Size 23.00 KB (23,552 bytes)  
 Version 5.2.3790.0 (srv03\_rtm.030324-2048)

[Ports]

[Serial]

Item Value  
 Name Communications Port (COM1)  
 Status OK  
 PNP Device ID ACPI\PNP0501\0  
 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  
 IRQ Channel IRQ 24  
 Memory Address 0xFF5E2000-0xFF5E2007  
 Driver c:\windows\system32\drivers\serial.sys (5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447), 191.00 KB (195,584 bytes), 3/25/2005 4:00 AM)

[Parallel]

Item Value

[Storage]

[Drives]

Item Value  
 Drive C:  
 Description Local Fixed Disk  
 Compressed No  
 File System NTFS  
 Size 33.27 GB (35,726,962,688 bytes)  
 Free Space 26.85 GB (28,825,997,312 bytes)

Volume Name  
 Volume Serial Number 033B5AD3

Drive D:  
 Description CD-ROM Disc

Drive G:  
 Description Local Fixed Disk  
 Compressed No  
 File System NTFS  
 Size 25.00 GB (26,841,444,352 bytes)

Free Space 12.37 GB (13,280,890,880 bytes)

Volume Name GDRIVE  
 Volume Serial Number COD345C7

Drive L:  
 Description Local Fixed Disk  
 Compressed Not Available  
 File System Not Available  
 Size Not Available  
 Free Space Not Available  
 Volume Name Not Available  
 Volume Serial Number Not Available

Drive Z:  
 Description Network Connection  
 Provider Name \\hpwieshare\wiew

[Disks]

Item Value  
 Description Disk drive  
 Manufacturer (Standard disk drives)  
 Model COMPAQ LOGICAL VOLUME SCSI  
 Disk Device  
 Bytes/Sector 512  
 Media Loaded Yes  
 Media Type Fixed hard disk  
 Partitions 3  
 SCSI Bus 0  
 SCSI Logical Unit 0  
 SCSI Port 12  
 SCSI Target ID 4  
 Sectors/Track 63  
 Size 33.88 GB (36,380,413,440 bytes)

Total Cylinders 4,423  
 Total Sectors 71,055,495  
 Total Tracks 1,127,865  
 Tracks/Cylinder 255  
 Partition Disk #39, Partition #0  
 Partition Size 100.00 MB (104,857,600 bytes)

Partition Starting Offset 17,408 bytes  
 Partition Disk #39, Partition #1  
 Partition Size 400.00 MB (419,430,400 bytes)

Partition Starting Offset 104,875,008 bytes

Partition Disk #39, Partition #2  
 Partition Size 33.27 GB (35,726,964,736 bytes)

Partition Starting Offset 658,523,648 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 2  
 SCSI Bus 0  
 SCSI Logical Unit 1  
 SCSI Port 2  
 SCSI Target ID 1  
 Sectors/Track 63  
 Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512  
 Total Sectors 329,525,280  
 Total Tracks 5,230,560  
 Tracks/Cylinder 255  
 Partition Disk #3, Partition #0  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #3, Partition #1  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3  
 SCSI Bus 0  
 SCSI Logical Unit 2  
 SCSI Port 2  
 SCSI Target ID 0  
 Sectors/Track 63  
 Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725  
 Total Sectors 1,328,977,125  
 Total Tracks 21,094,875  
 Tracks/Cylinder 255  
 Partition Disk #4, Partition #0  
 Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #4, Partition #1  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes

Partition Disk #4, Partition #2  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 2  
 SCSI Bus 0  
 SCSI Logical Unit 1  
 SCSI Port 2  
 SCSI Target ID 1  
 Sectors/Track 63  
 Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512  
 Total Sectors 329,525,280  
 Total Tracks 5,230,560  
 Tracks/Cylinder 255  
 Partition Disk #5, Partition #0  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #5, Partition #1  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3  
 SCSI Bus 0  
 SCSI Logical Unit 2  
 SCSI Port 2  
 SCSI Target ID 1  
 Sectors/Track 63  
 Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725  
 Total Sectors 1,328,977,125  
 Total Tracks 21,094,875  
 Tracks/Cylinder 255  
 Partition Disk #6, Partition #0  
 Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #6, Partition #1  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes

Partition Disk #6, Partition #2  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 2  
 SCSI Bus 0

SCSI Logical Unit 1  
 SCSI Target ID 0  
 Sectors/Track 63  
 Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512  
 Total Sectors 329,525,280  
 Total Tracks 5,230,560  
 Tracks/Cylinder 255  
 Partition Disk #15, Partition #0  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #15, Partition #1  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3  
 SCSI Bus 0  
 SCSI Logical Unit 2  
 SCSI Port 5  
 SCSI Target ID 0

Sectors/Track 63  
 Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725  
 Total Sectors 1,328,977,125  
 Total Tracks 21,094,875  
 Tracks/Cylinder 255  
 Partition Disk #16, Partition #0  
 Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #16, Partition #1  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes

Partition Disk #16, Partition #2  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 2  
 SCSI Bus 0  
 SCSI Logical Unit 1  
 SCSI Port 8  
 SCSI Target ID 0  
 Sectors/Track 63  
 Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512  
 Total Sectors 329,525,280  
 Total Tracks 5,230,560

Tracks/Cylinder 255  
Partition Disk #25, Partition #0  
Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #25, Partition #1  
Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3  
SCSI Bus 0  
SCSI Logical Unit 2  
SCSI Port 8  
SCSI Target ID 0  
Sectors/Track 63  
Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725  
Total Sectors 1,328,977,125  
Total Tracks 21,094,875  
Tracks/Cylinder 255  
Partition Disk #26, Partition #0  
Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #26, Partition #1  
Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes

Partition Disk #26, Partition #2  
Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 2  
SCSI Bus 0  
SCSI Logical Unit 1  
SCSI Port 10  
SCSI Target ID 0  
Sectors/Track 63  
Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512  
Total Sectors 329,525,280  
Total Tracks 5,230,560  
Tracks/Cylinder 255  
Partition Disk #31, Partition #0  
Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #31, Partition #1

Partition Size 56.00 GB (60,129,542,144 bytes)  
Partition Starting Offset 108,584,239,104 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 3  
SCSI Bus 0  
SCSI Logical Unit 2  
SCSI Port 10  
SCSI Target ID 0  
Sectors/Track 63  
Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725  
Total Sectors 1,328,977,125  
Total Tracks 21,094,875  
Tracks/Cylinder 255  
Partition Disk #32, Partition #0  
Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #32, Partition #1  
Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes

Partition Disk #32, Partition #2  
Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 2  
SCSI Bus 0  
SCSI Logical Unit 1  
SCSI Port 10  
SCSI Target ID 1  
Sectors/Track 63  
Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512  
Total Sectors 329,525,280  
Total Tracks 5,230,560  
Tracks/Cylinder 255  
Partition Disk #33, Partition #0  
Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #33, Partition #1  
Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 bytes

Description Disk drive  
Manufacturer (Standard disk drives)  
Model COMPAQ MSA1000 VOLUME SCSI  
Disk Device

Bytes/Sector 512 Yes  
Media Type Fixed hard disk  
Partitions 3  
SCSI Bus 0  
SCSI Logical Unit 2  
SCSI Port 10  
SCSI Target ID 1  
Sectors/Track 63  
Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725  
Total Sectors 1,328,977,125  
Total Tracks 21,094,875  
Tracks/Cylinder 255  
Partition Disk #34, Partition #0  
Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #34, Partition #1  
Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes

Partition Disk #34, Partition #2  
Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 2  
SCSI Bus 0

SCSI Logical Unit 1  
SCSI Port 4  
SCSI Target ID 0  
Sectors/Track 63  
Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512  
Total Sectors 329,525,280  
Total Tracks 5,230,560  
Tracks/Cylinder 255  
Partition Disk #11, Partition #0  
Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #11, Partition #1  
Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3  
SCSI Bus 0  
SCSI Logical Unit 2  
SCSI Port 4  
SCSI Target ID 0  
Sectors/Track 63

Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725

Total Sectors 1,328,977,125

Total Tracks 21,094,875

Tracks/Cylinder 255

Partition Disk #12, Partition #0

Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #12, Partition #1

Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes

Partition Disk #12, Partition #2

Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 2

SCSI Bus 0

SCSI Logical Unit 1

SCSI Port 4

SCSI Target ID 1

Sectors/Track 63

Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512

Total Sectors 329,525,280

Total Tracks 5,230,560

Tracks/Cylinder 255

Partition Disk #13, Partition #0

Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #13, Partition #1

Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3

SCSI Bus 0

SCSI Logical Unit 2

SCSI Port 4

SCSI Target ID 1

Sectors/Track 63

Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725

Total Sectors 1,328,977,125

Total Tracks 21,094,875

Tracks/Cylinder 255

Partition Disk #14, Partition #0

Partition Size 320.00 GB (343,597,383,680 bytes)  
Partition Starting Offset 136,314,880 bytes

Partition Disk #14, Partition #1

Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes

Partition Disk #14, Partition #2

Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 2

SCSI Bus 0

SCSI Logical Unit 1

SCSI Port 3

SCSI Target ID 0

Sectors/Track 63

Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512

Total Sectors 329,525,280

Total Tracks 5,230,560

Tracks/Cylinder 255

Partition Disk #7, Partition #0

Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #7, Partition #1

Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3

SCSI Bus 0

SCSI Logical Unit 2

SCSI Port 3

SCSI Target ID 0

Sectors/Track 63

Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725

Total Sectors 1,328,977,125

Total Tracks 21,094,875

Tracks/Cylinder 255

Partition Disk #8, Partition #0

Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #8, Partition #1

Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes

Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 2

SCSI Bus 0

SCSI Logical Unit 1

SCSI Port 3

SCSI Target ID 1

Sectors/Track 63

Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512

Total Sectors 329,525,280

Total Tracks 5,230,560

Tracks/Cylinder 255

Partition Disk #9, Partition #0

Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #9, Partition #1

Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3

SCSI Bus 0

SCSI Logical Unit 2

SCSI Port 3

SCSI Target ID 1

Sectors/Track 63

Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725

Total Sectors 1,328,977,125

Total Tracks 21,094,875

Tracks/Cylinder 255

Partition Disk #10, Partition #0

Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #10, Partition #1

Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes

Partition Disk #10, Partition #2

Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 2  
 SCSI Bus 0  
 SCSI Logical Unit 1  
 SCSI Port 7  
 SCSI Target ID 0  
 Sectors/Track 63  
 Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512  
 Total Sectors 329,525,280  
 Total Tracks 5,230,560  
 Tracks/Cylinder 255  
 Partition Disk #21, Partition #0  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #21, Partition #1  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3  
 SCSI Bus 0  
 SCSI Logical Unit 2  
 SCSI Port 7  
 SCSI Target ID 0  
 Sectors/Track 63  
 Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725  
 Total Sectors 1,328,977,125  
 Total Tracks 21,094,875  
 Tracks/Cylinder 255  
 Partition Disk #22, Partition #0  
 Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #22, Partition #1  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes  
 Partition Disk #22, Partition #2  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 2  
 SCSI Bus 0  
 SCSI Logical Unit 1

SCSI Target ID 1  
 Sectors/Track 63  
 Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512  
 Total Sectors 329,525,280  
 Total Tracks 5,230,560  
 Tracks/Cylinder 255  
 Partition Disk #23, Partition #0  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #23, Partition #1  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3  
 SCSI Bus 0  
 SCSI Logical Unit 2  
 SCSI Port 7  
 SCSI Target ID 1  
 Sectors/Track 63  
 Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725  
 Total Sectors 1,328,977,125  
 Total Tracks 21,094,875  
 Tracks/Cylinder 255  
 Partition Disk #24, Partition #0  
 Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #24, Partition #1  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes

Partition Disk #24, Partition #2  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 2  
 SCSI Bus 0  
 SCSI Logical Unit 1  
 SCSI Port 0  
 SCSI Target ID 0  
 Sectors/Track 63  
 Size 1.09 TB (1,199,994,324,480 bytes)

Total Cylinders 145,891  
 Total Sectors 2,343,738,915  
 Total Tracks 37,202,205  
 Tracks/Cylinder 255

Partition Size 101.00 GB (108,447,924,224 bytes)  
 Partition Starting Offset 135,266,304 bytes

Partition Disk #2, Partition #1  
 Partition Size 25.00 GB (26,841,448,448 bytes)

Partition Starting Offset 1,173,155,217,408 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 2  
 SCSI Bus 0  
 SCSI Logical Unit 1  
 SCSI Port 6  
 SCSI Target ID 0  
 Sectors/Track 63  
 Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512  
 Total Sectors 329,525,280  
 Total Tracks 5,230,560  
 Tracks/Cylinder 255  
 Partition Disk #17, Partition #0  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #17, Partition #1  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3  
 SCSI Bus 0  
 SCSI Logical Unit 2  
 SCSI Port 6  
 SCSI Target ID 0

Sectors/Track 63  
 Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725  
 Total Sectors 1,328,977,125  
 Total Tracks 21,094,875  
 Tracks/Cylinder 255  
 Partition Disk #18, Partition #0  
 Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #18, Partition #1  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes

Partition Disk #18, Partition #2  
 Partition Size 56.00 GB (60,129,542,144 bytes)



Partition Starting Offset 452,181,622,784 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 2  
 SCSI Bus 0  
 SCSI Logical Unit 1  
 SCSI Port 6  
 SCSI Target ID 1  
 Sectors/Track 63  
 Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512  
 Total Sectors 329,525,280  
 Total Tracks 5,230,560  
 Tracks/Cylinder 255  
 Partition Disk #19, Partition #0  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #19, Partition #1  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3  
 SCSI Bus 0  
 SCSI Logical Unit 2  
 SCSI Port 6  
 SCSI Target ID 1  
 Sectors/Track 63  
 Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725  
 Total Sectors 1,328,977,125  
 Total Tracks 21,094,875  
 Tracks/Cylinder 255  
 Partition Disk #20, Partition #0  
 Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #20, Partition #1  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes  
 Partition Disk #20, Partition #2  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 bytes

Description Disk drive  
 Manufacturer (Standard disk drives)  
 Model COMPAQ MSA1000 VOLUME SCSI  
 Disk Device  
 Bytes/Sector 512

Media Type Fixed hard disk  
 Partitions 2  
 SCSI Bus 0  
 SCSI Logical Unit 1  
 SCSI Port 9  
 SCSI Target ID 0  
 Sectors/Track 63  
 Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512  
 Total Sectors 329,525,280  
 Total Tracks 5,230,560  
 Tracks/Cylinder 255  
 Partition Disk #27, Partition #0  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #27, Partition #1  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3  
 SCSI Bus 0  
 SCSI Logical Unit 2  
 SCSI Port 9  
 SCSI Target ID 0  
 Sectors/Track 63  
 Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725  
 Total Sectors 1,328,977,125  
 Total Tracks 21,094,875  
 Tracks/Cylinder 255  
 Partition Disk #28, Partition #0  
 Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #28, Partition #1  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes  
 Partition Disk #28, Partition #2  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 2  
 SCSI Bus 0  
 SCSI Logical Unit 1  
 SCSI Port 9  
 SCSI Target ID 1  
 Sectors/Track 63

Size 157.13 GB (168,716,943,360 bytes)  
 Total Cylinders 20,512  
 Total Sectors 329,525,280  
 Total Tracks 5,230,560  
 Tracks/Cylinder 255  
 Partition Disk #29, Partition #0  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #29, Partition #1  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3  
 SCSI Bus 0  
 SCSI Logical Unit 2  
 SCSI Port 9  
 SCSI Target ID 1  
 Sectors/Track 63  
 Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725  
 Total Sectors 1,328,977,125  
 Total Tracks 21,094,875  
 Tracks/Cylinder 255  
 Partition Disk #30, Partition #0  
 Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #30, Partition #1  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes

Partition Disk #30, Partition #2  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 2  
 SCSI Bus 0

SCSI Logical Unit 1  
 SCSI Port 11  
 SCSI Target ID 0  
 Sectors/Track 63  
 Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512  
 Total Sectors 329,525,280  
 Total Tracks 5,230,560  
 Tracks/Cylinder 255  
 Partition Disk #35, Partition #0  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #35, Partition #1  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3  
 SCSI Bus 0  
 SCSI Logical Unit 2  
 SCSI Port 11  
 SCSI Target ID 0  
 Sectors/Track 63  
 Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725  
 Total Sectors 1,328,977,125  
 Total Tracks 21,094,875  
 Tracks/Cylinder 255  
 Partition Disk #36, Partition #0  
 Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #36, Partition #1  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes  
 Partition Disk #36, Partition #2  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 2  
 SCSI Bus 0  
 SCSI Logical Unit 1  
 SCSI Port 11  
 SCSI Target ID 1  
 Sectors/Track 63  
 Size 157.13 GB (168,716,943,360 bytes)

Total Cylinders 20,512  
 Total Sectors 329,525,280  
 Total Tracks 5,230,560  
 Tracks/Cylinder 255

Partition Disk #37, Partition #0  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #37, Partition #1  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 108,584,239,104 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 3  
 SCSI Bus 0  
 SCSI Logical Unit 2  
 SCSI Port 11  
 SCSI Target ID 1  
 Sectors/Track 63  
 Size 633.71 GB (680,436,288,000 bytes)

Total Cylinders 82,725  
 Total Sectors 1,328,977,125  
 Total Tracks 21,094,875  
 Tracks/Cylinder 255  
 Partition Disk #38, Partition #0  
 Partition Size 320.00 GB (343,597,383,680 bytes)

Partition Starting Offset 136,314,880 bytes

Partition Disk #38, Partition #1  
 Partition Size 101.00 GB (108,447,924,224 bytes)

Partition Starting Offset 343,733,698,560 bytes  
 Partition Disk #38, Partition #2  
 Partition Size 56.00 GB (60,129,542,144 bytes)

Partition Starting Offset 452,181,622,784 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 0  
 SCSI Logical Unit 1  
 SCSI Port 1  
 SCSI Target ID 0  
 Sectors/Track 63  
 Size 1.91 TB (2,099,988,011,520 bytes)

Total Cylinders 255,309  
 Total Sectors 4,101,539,085  
 Total Tracks 65,103,795  
 Tracks/Cylinder 255  
 Partition Disk #0, Partition #0  
 Partition Size 1.91 TB (2,099,860,731,392 bytes)

Partition Starting Offset 135,266,304 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 0  
 SCSI Logical Unit 1  
 SCSI Port 1  
 SCSI Target ID 1  
 Sectors/Track 63

Size 1.91 TB (2,099,988,011,520 bytes)  
 Total Cylinders 255,309  
 Total Sectors 4,101,539,085  
 Total Tracks 65,103,795  
 Tracks/Cylinder 255  
 Partition Disk #1, Partition #0  
 Partition Size 1.91 TB (2,099,860,731,392 bytes)

Partition Starting Offset 135,266,304 bytes

[SCSI]

Item Value  
 Name Emulex LightPulse LP1050, Storport  
 Miniport Driver  
 Manufacturer Emulex  
 Status OK  
 PNP Device ID  
 PCI\VEN\_10DF&DEV\_F0A5&SUBSYS\_S\_F0A510DF&REV\_01\4&2C178B65&0&08

Memory Address 0x90003000-0x90003FFF  
 Memory Address 0x90002000-0x900020FF  
 I/O Port 0x00002100-0x000021FF  
 IRQ Channel IRQ 27  
 Driver  
 c:\windows\system32\drivers\elxstor.sys (6-1.11X1 11/07/2005 WS2K3 64 bit IA64 built by: WinDDK, 777.50 KB (796,160 bytes), 6/24/2006 6:26 PM)

Name Emulex LightPulse LP1050, Storport  
 Miniport Driver  
 Manufacturer Emulex  
 Status OK  
 PNP Device ID  
 PCI\VEN\_10DF&DEV\_F0A5&SUBSYS\_S\_F0A510DF&REV\_01\4&2C178B65&0&09

Memory Address 0x90001000-0x90001FFF  
 Memory Address 0x90000000-0x9FFFFFFF  
 I/O Port 0x00002000-0x00003FFF  
 IRQ Channel IRQ 28  
 Driver  
 c:\windows\system32\drivers\elxstor.sys (6-1.11X1 11/07/2005 WS2K3 64 bit IA64 built by: WinDDK, 777.50 KB (796,160 bytes), 6/24/2006 6:26 PM)

Name QLogic Fibre Channel Adapter

Manufacturer QLogic  
 Status OK  
 PNP Device ID  
 PCI\VEN\_1077&DEV\_2312&SUBSYS\_01010E11&REV\_02\4&915E9088&0&08  
 I/O Port 0x00004100-0x000041FF  
 Memory Address 0xA0041000-0xA0041FFF  
 IRQ Channel IRQ 38

Driver  
 c:\windows\system32\drivers\ql2300.sys (8.2.3.11 (wia64 VI), 655.00 KB (670,720 bytes), 6/24/2006 6:39 PM)

Name QLogic Fibre Channel Adapter  
 Manufacturer QLogic

```

Status      OK
PNP Device ID
    PCI\VEN_1077&DEV_2312&SUBSYS
    _01010E11&REV_02\4&915E908&0&09
I/O Port    0x00004000-0x00005FFF
Memory Address      0xA0040000-
0xA0040FFF
IRQ Channel IRQ 39
Driver
    c:\windows\system32\drivers\ql230
0.sys (8.2.3.11 (wia64 VI), 655.00 KB (670,720
bytes), 6/24/2006 6:39 PM)

Name        QLogic Fibre Channel Adapter

Manufacturer QLogic
Status      OK
PNP Device ID
    PCI\VEN_1077&DEV_2312&SUBSYS
    _01010E11&REV_02\4&19EBB955&0&08

I/O Port    0x00006100-0x000061FF
Memory Address      0xB0041000-
0xB0041FFF
IRQ Channel IRQ 49
Driver
    c:\windows\system32\drivers\ql230
0.sys (8.2.3.11 (wia64 VI), 655.00 KB (670,720
bytes), 6/24/2006 6:39 PM)

Name        QLogic Fibre Channel Adapter

Manufacturer QLogic
Status      OK
PNP Device ID
    PCI\VEN_1077&DEV_2312&SUBSYS
    _01010E11&REV_02\4&19EBB955&0&09

I/O Port    0x00006000-0x00007FFF
Memory Address      0xB0040000-
0xB0040FFF
IRQ Channel IRQ 50
Driver
    c:\windows\system32\drivers\ql230
0.sys (8.2.3.11 (wia64 VI), 655.00 KB (670,720
bytes), 6/24/2006 6:39 PM)

Name        Smart Array P600 Controller

Manufacturer Hewlett-Packard Company
Status      OK
PNP Device ID
    PCI\VEN_103C&DEV_3220&SUBSYS
    _3225103C&REV_00\4&15291AB&0&08
Memory Address      0xC01C0000-
0xC01C1FFF
I/O Port    0x00008000-0x00009FFF
Memory Address      0xC0140000-
0xC017FFFF
IRQ Channel IRQ 60
Driver
    c:\windows\system32\drivers\hpcis
ss2.sys (5.6.0.64 Build 4 (ia64) built by:
WINBUILD1, 89.00 KB (91,136 bytes), 9/1/2005
3:05 AM)

Name        QLogic Fibre Channel Adapter

Manufacturer QLogic

Status      OK
PNP Device ID
    PCI\VEN_1077&DEV_2312&SUBSYS
    _01010E11&REV_02\4&24543408&0&08

```

```

Memory Address 0x0000A100-0x0000A1FF
0xD0041FFF
IRQ Channel IRQ 71
Driver
    c:\windows\system32\drivers\ql230
0.sys (8.2.3.11 (wia64 VI), 655.00 KB (670,720
bytes), 6/24/2006 6:39 PM)

Name        QLogic Fibre Channel Adapter

Manufacturer QLogic
Status      OK
PNP Device ID
    PCI\VEN_1077&DEV_2312&SUBSYS
    _01010E11&REV_02\4&24543408&0&09
I/O Port    0x0000A000-0x0000BFFF
Memory Address      0xD0040000-
0xD0040FFF
IRQ Channel IRQ 72
Driver
    c:\windows\system32\drivers\ql230
0.sys (8.2.3.11 (wia64 VI), 655.00 KB (670,720
bytes), 6/24/2006 6:39 PM)

Name        QLogic Fibre Channel Adapter

Manufacturer QLogic
Status      OK
PNP Device ID
    PCI\VEN_1077&DEV_2312&SUBSYS
    _01010E11&REV_02\4&5D9CB86&0&08
I/O Port    0x0000C100-0x0000C1FF
Memory Address      0xE0041000-
0xE0041FFF
IRQ Channel IRQ 82
Driver
    c:\windows\system32\drivers\ql230
0.sys (8.2.3.11 (wia64 VI), 655.00 KB (670,720
bytes), 6/24/2006 6:39 PM)

Name        QLogic Fibre Channel Adapter

Manufacturer QLogic
Status      OK
PNP Device ID
    PCI\VEN_1077&DEV_2312&SUBSYS
    _01010E11&REV_02\4&5D9CB86&0&09
I/O Port    0x0000C000-0x0000DFFF
Memory Address      0xE0040000-
0xE0040FFF
IRQ Channel IRQ 83
Driver
    c:\windows\system32\drivers\ql230
0.sys (8.2.3.11 (wia64 VI), 655.00 KB (670,720
bytes), 6/24/2006 6:39 PM)

Name        QLogic Fibre Channel Adapter

Manufacturer QLogic
Status      OK
PNP Device ID
    PCI\VEN_1077&DEV_2312&SUBSYS
    _01010E11&REV_02\4&1E72F330&0&08
I/O Port    0x0000E100-0x0000E1FF
Memory Address      0xF0041000-
0xF0041FFF
IRQ Channel IRQ 93
Driver
    c:\windows\system32\drivers\ql230
0.sys (8.2.3.11 (wia64 VI), 655.00 KB (670,720
bytes), 6/24/2006 6:39 PM)

```

```

Name        QLogic Fibre Channel Adapter
Manufacturer QLogic
Status      OK
PNP Device ID
    PCI\VEN_1077&DEV_2312&SUBSYS
    _01010E11&REV_02\4&1E72F330&0&09
I/O Port    0x0000E000-0x0000FFFF
Memory Address      0xF0040000-
0xF0040FFF
IRQ Channel IRQ 94
Driver
    c:\windows\system32\drivers\ql230
0.sys (8.2.3.11 (wia64 VI), 655.00 KB (670,720
bytes), 6/24/2006 6:39 PM)

[IDE]

Item        Value

[Printing]

Name        Driver        Port Name    Server
Name

[Problem Devices]

Device      PNP Device ID    Error
Code

[USB]

Device      PNP Device ID
NEC PCI to USB Open Host Controller
    PCI\VEN_1033&DEV_0035&SUBSYS
    _022603F0&REV_43\4&4F5EBC7&0&10
NEC PCI to USB Open Host Controller
    PCI\VEN_1033&DEV_0035&SUBSYS
    _022603F0&REV_43\4&4F5EBC7&0&11
Standard Enhanced PCI to USB Host Controller
    PCI\VEN_1033&DEV_00E0&SUBSYS
    _032603F0&REV_04\4&4F5EBC7&0&12

[Software Environment]

[System Drivers]

Name        Description    File        Type
Started    Start Mode    State
Status      Error Control Accept

Pause
abiosdsk    Accept Stop
Driver      Abiosdsk      Not Available Kernel
No          Disabled      Stopped
OK          Ignore        No

acpi        Microsoft ACPI Driver
c:\windows\system32\drivers\acpi.s
ys          Kernel Driver Yes
Running    OK            Normal
No         Yes

acpiec     ACPIEC
c:\windows\system32\drivers\acpie
c.sys      Kernel Driver No
Stopped   OK            Normal
No        No

adpu160m   adpu160m
c:\windows\system32\drivers\adpu
160m.sys   Kernel Driver No
Stopped   OK            Normal
No        No

```

adpu320 adpu320  
 c:\windows\system32\drivers\adpu  
 320.sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

afcnt afcnt  
 c:\windows\system32\drivers\afcnt.  
 sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

afd AFD  
 c:\windows\system32\drivers\afd.sy  
 s Kernel Driver Yes System  
 Running OK Normal  
 No Yes

agp460 Intel AGP Bus Filter  
 60.sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

aic78u2 aic78u2  
 c:\windows\system32\drivers\aic78  
 u2.sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

aic78xx aic78xx  
 c:\windows\system32\drivers\aic78  
 xx.sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

aliide AliIde  
 c:\windows\system32\drivers\aliide.  
 sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

arc arc Not Available Kernel  
 Driver No Disabled Stopped  
 OK Normal No

asynmac RAS Asynchronous Media Driver  
 c:\windows\system32\drivers\asyn  
 mac.sys Kernel Driver No Manual  
 Stopped OK Normal  
 No No

atapi Standard IDE/ESDI Hard Disk  
 Controller  
 c:\windows\system32\drivers\atapi.  
 sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

atdisk Atdisk Not Available Kernel  
 Driver No Disabled Stopped  
 OK Ignore No

ati2mtag ati2mtag  
 c:\windows\system32\drivers\ati2m  
 tag.sys Kernel Driver Yes Manual  
 Running OK Ignore  
 No Yes

atmarpc ATM ARP Client Protocol  
 c:\windows\system32\drivers\atmar  
 pc.sys Kernel Driver No Manual  
 Stopped OK Normal  
 No No

audstub Audio Stub Driver  
 c:\windows\system32\drivers\audst  
 ub.sys Kernel Driver Yes Manual  
 Running OK Normal  
 No Yes

b57nd Broadcom NetXtreme Gigabit  
 Ethernet  
 c:\windows\system32\drivers\b57x  
 p64.sys Kernel Driver No Manual  
 Stopped OK Normal  
 No No

beep Beep  
 c:\windows\system32\drivers\beep.  
 sys Kernel Driver Yes System  
 Running OK Normal  
 No Yes

cbidf cbidf  
 c:\windows\system32\drivers\cbidf2  
 k.sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

cbidf2k cbidf2k  
 c:\windows\system32\drivers\cbidf2  
 k.sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

cdfs Cdfs  
 c:\windows\system32\drivers\cdfs.s  
 ys File System Driver Yes  
 Disabled Running OK  
 Normal No Yes

cdrom CD-ROM Driver  
 c:\windows\system32\drivers\cdro  
 m.sys Kernel Driver Yes System  
 Running OK Normal  
 No Yes

changer Changer Not Available Kernel  
 Driver No System Stopped  
 OK Ignore No

clusdisk Cluster Disk Driver  
 c:\windows\system32\drivers\clusdi  
 sk.sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

cmdide CmdIde  
 c:\windows\system32\drivers\cmdid  
 e.sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

cpqarry2 cpqarry2  
 c:\windows\system32\drivers\cpqar  
 ry2.sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

cpqcisse CPQCISSE  
 c:\windows\system32\drivers\cpqci  
 sse.sys Kernel Driver No Manual  
 Stopped OK Normal  
 No No

cpqcissm cpqcissm  
 c:\windows\system32\drivers\cpqci  
 ssm.sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

cpqfcac CPQFCAC  
 c:\windows\system32\drivers\cpqfc  
 ac.sys Kernel Driver Yes Boot  
 Running OK Normal  
 No Yes

cpqfcalm cpqfcalm  
 c:\windows\system32\drivers\cpqfc  
 alm.sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

cpqteam HP Network Configuration Utility  
 c:\windows\system32\drivers\cpqte  
 am.sys Kernel Driver No Manual  
 Stopped OK Normal  
 No No

crdisk CRC Disk Filter Driver  
 c:\windows\system32\drivers\crckis  
 k.sys Kernel Driver Yes Boot  
 Running OK Normal  
 No Yes

dfsdriver DfsDriver  
 c:\windows\system32\drivers\dfs.sy  
 s File System Driver Yes  
 Boot Running OK  
 Normal No Yes

disk Disk Driver  
 c:\windows\system32\drivers\disk.s  
 ys Kernel Driver Yes Boot  
 Running OK Normal  
 No Yes

dmboot dmboot  
 c:\windows\system32\drivers\dmbo  
 ot.sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

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\dmlo  
 ad.sys Kernel Driver Yes Boot  
 Running OK Normal  
 No Yes

dpti2o dpti2o  
 c:\windows\system32\drivers\dpti2  
 o.sys Kernel Driver No Disabled  
 Stopped OK Normal  
 No No

e1000 Intel(R) PRO/1000 Network  
 Connection Driver  
 c:\windows\system32\drivers\e100  
 0645.sys Kernel Driver Yes Manual  
 Running OK Normal  
 No Yes

elxadjct elxadjct  
 c:\windows\system32\drivers\elxadj  
 ct.sys Kernel Driver Yes Boot  
 Running OK Normal  
 No Yes

elxstor elxstor  
 c:\windows\system32\drivers\elxsto  
 r.sys Kernel Driver Yes Boot  
 Running OK Normal  
 No Yes

fastfat Fastfat  
 c:\windows\system32\drivers\fastfa  
 t.sys File System Driver No  
 Disabled Stopped OK  
 Normal No No

fdc Fdc  
 c:\windows\system32\drivers\fdc.sy  
 s Kernel Driver No System  
 Stopped OK Ignore  
 No No

fips Fips  
 c:\windows\system32\drivers\fips.s  
 ys Kernel Driver Yes System  
 Running OK Normal  
 No Yes

|          |                                     |               |         |
|----------|-------------------------------------|---------------|---------|
| flpydisk | Flpydisk                            |               |         |
|          | c:\windows\system32\drivers\flpydi  |               |         |
| sk.sys   | Kernel Driver No                    | System        |         |
|          | Stopped OK                          | Ignore        |         |
|          | No No                               |               |         |
| ftmgr    | FitMgr                              |               |         |
|          | c:\windows\system32\drivers\ftmgr   |               |         |
| .sys     | File System Driver                  | Yes           |         |
|          | Boot Running                        | OK            |         |
|          | Normal No                           | Yes           |         |
| 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\msgp    |               |         |
| c.sys    | Kernel Driver Yes                   | Manual        |         |
|          | Running OK                          | Normal        |         |
|          | No Yes                              |               |         |
| hidusb   | Microsoft HID Class Driver          |               |         |
|          | c:\windows\system32\drivers\hidus   |               |         |
| b.sys    | Kernel Driver Yes                   | Manual        |         |
|          | Running OK                          | Ignore        |         |
|          | No Yes                              |               |         |
| hpcisss  | hpcisss                             |               |         |
|          | c:\windows\system32\drivers\hpcis   |               |         |
| ss.sys   | Kernel Driver Yes                   | Boot          |         |
|          | Running OK                          | Normal        |         |
|          | No Yes                              |               |         |
| hpcisss2 | HpCISs2                             |               |         |
|          | c:\windows\system32\drivers\hpcis   |               |         |
| ss2.sys  | Kernel Driver Yes                   | Boot          |         |
|          | Running OK                          | Normal        |         |
|          | No Yes                              |               |         |
| hphlth   | HP Baseboard Management             |               |         |
|          | Controller Interface Device         |               |         |
|          | c:\windows\system32\drivers\hphlth  |               |         |
| h.sys    | Kernel Driver Yes                   | Manual        |         |
|          | Running OK                          | Normal        |         |
|          | No Yes                              |               |         |
| hmpmpser | HP MP Driver                        |               |         |
|          | c:\windows\system32\drivers\hmp     |               |         |
| ser.sys  | Kernel Driver No                    | Manual        |         |
|          | Stopped OK                          | Normal        |         |
|          | No No                               |               |         |
| hpn      | hpn                                 |               |         |
|          | c:\windows\system32\drivers\hpn.s   |               |         |
| ys       | Kernel Driver No                    | Disabled      |         |
|          | Stopped OK                          | Normal        |         |
|          | No No                               |               |         |
| http     | HTTP                                |               |         |
|          | c:\windows\system32\drivers\http.s  |               |         |
| ys       | Kernel Driver No                    | Manual        |         |
|          | Stopped OK                          | Normal        |         |
|          | No No                               |               |         |
| i2omgmt  | i2omgmt                             | Not Available | Kernel  |
| Driver   | No                                  | System        | Stopped |
|          | OK                                  | Normal        | No      |
|          | No                                  |               |         |
| iirsp    | iirsp                               | Not Available | Kernel  |
| Driver   | No                                  | Disabled      | 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\intelid |          |  |
| e.sys          | Kernel Driver No                    | Disabled |  |
|                | Stopped OK                          | Normal   |  |
|                | No No                               |          |  |
| ip6fw          | IPV6 Windows Firewall Driver        |          |  |
|                | c:\windows\system32\drivers\ip6fw   |          |  |
| .sys           | Kernel Driver No                    | Manual   |  |
|                | Stopped OK                          | Normal   |  |
|                | No No                               |          |  |
| ipfilterdriver | IP Traffic Filter Driver            |          |  |
|                | c:\windows\system32\drivers\ipftdr  |          |  |
| v.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\isapn   |          |  |
| p.sys          | Kernel Driver No                    | Disabled |  |
|                | Stopped OK                          | Critical |  |
|                | No No                               |          |  |
| kbdclass       | Keyboard Class Driver               |          |  |
|                | c:\windows\system32\drivers\kbddl   |          |  |
| ass.sys        | Kernel Driver Yes                   | System   |  |
|                | Running OK                          | Normal   |  |
|                | No Yes                              |          |  |
| kbdhid         | Keyboard HID Driver                 |          |  |
|                | c:\windows\system32\drivers\kbdhi   |          |  |
| d.sys          | Kernel Driver Yes                   | System   |  |
|                | Running OK                          | Ignore   |  |
|                | No Yes                              |          |  |
| ksecdd         | KSecDD                              |          |  |
|                | c:\windows\system32\drivers\ksecd   |          |  |
| d.sys          | Kernel Driver Yes                   | Boot     |  |
|                | Running OK                          | Normal   |  |
|                | No Yes                              |          |  |
| ksthunk        | Kernel Streaming WOW64 Thunk        |          |  |
|                | Service                             |          |  |
|                | c:\windows\system32\drivers\ksth    |          |  |
| nk.sys         | Kernel Driver Yes                   | Manual   |  |
|                | Running OK                          | Normal   |  |
|                | No Yes                              |          |  |
| lp6nds35       | lp6nds35                            |          |  |
|                | c:\windows\system32\drivers\lp6nd   |          |  |
| s35.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\lpxnd   |          |  |
| s.sys          | Kernel Driver Yes                   | Boot     |  |
|                | Running OK                          | Normal   |  |
|                | No Yes                              |          |  |

|          |                                    |               |         |
|----------|------------------------------------|---------------|---------|
| mnmdd    | mnmdd                              | Not Available | Kernel  |
| Driver   | No                                 | System        | Stopped |
|          | OK                                 | Ignore        | No      |
| modem    | Modem                              |               |         |
|          | c:\windows\system32\drivers\mode   |               |         |
| m.sys    | Kernel Driver No                   | Manual        |         |
|          | Stopped OK                         | Ignore        |         |
|          | No No                              |               |         |
| mouclass | Mouse Class Driver                 |               |         |
|          | c:\windows\system32\drivers\moucl  |               |         |
| ass.sys  | Kernel Driver Yes                  | System        |         |
|          | Running OK                         | Normal        |         |
|          | No Yes                             |               |         |
| mouhid   | Mouse HID Driver                   |               |         |
|          | c:\windows\system32\drivers\mouh   |               |         |
| id.sys   | Kernel Driver Yes                  | Manual        |         |
|          | Running OK                         | Ignore        |         |
|          | No Yes                             |               |         |
| mountmgr | Mount Point Manager                |               |         |
|          | c:\windows\system32\drivers\moun   |               |         |
| tmgr.sys | Kernel Driver Yes                  | Boot          |         |
|          | Running OK                         | Normal        |         |
|          | No Yes                             |               |         |
| mraid35x | mraid35x                           |               |         |
|          | c:\windows\system32\drivers\mraid  |               |         |
| 35x.sys  | Kernel Driver No                   | Disabled      |         |
|          | Stopped OK                         | Normal        |         |
|          | No No                              |               |         |
| mrxdav   | WebDav Client Redirector           |               |         |
|          | c:\windows\system32\drivers\mrxd   |               |         |
| av.sys   | File System Driver                 | No            |         |
|          | Manual Stopped                     | OK            |         |
|          | Normal No                          | No            |         |
| mrxsmb   | MRXSMB                             |               |         |
|          | c:\windows\system32\drivers\mrxs   |               |         |
| mb.sys   | File System Driver                 | Yes           |         |
|          | System Running                     | OK            |         |
|          | Normal No                          | Yes           |         |
| msfs     | Msfs                               |               |         |
|          | c:\windows\system32\drivers\msfs.  |               |         |
| sys      | File System Driver                 | Yes           |         |
|          | System Running                     | OK            |         |
|          | Normal No                          | Yes           |         |
| mssmbios | Microsoft System Management        |               |         |
|          | BIOS Driver                        |               |         |
|          | c:\windows\system32\drivers\mssm   |               |         |
| bios.sys | Kernel Driver Yes                  | Manual        |         |
|          | 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.s |               |         |
| ys       | Kernel Driver Yes                  | Boot          |         |
|          | Running OK                         | Normal        |         |
|          | No Yes                             |               |         |
| ndistapi | Remote Access NDIS TAPI Driver     |               |         |
|          | c:\windows\system32\drivers\ndista |               |         |
| pi.sys   | Kernel Driver Yes                  | Manual        |         |
|          | Running OK                         | Normal        |         |
|          | No Yes                             |               |         |
| ndisuio  | NDIS Usermode I/O Protocol         |               |         |
|          | c:\windows\system32\drivers\ndisui |               |         |
| o.sys    | Kernel Driver No                   | Manual        |         |
|          | Stopped OK                         | Normal        |         |
|          | No No                              |               |         |

|         |   |                    |         |          |
|---------|---|--------------------|---------|----------|
| ndiswan | Remote Access NDIS WAN Driver           |                    |         |          |
| an.sys  | c:\windows\system32\drivers\ndiswan.sys | Kernel Driver      | Yes     | Manual   |
|         |   | Running            | OK      | Normal   |
|         |   | No                 | Yes     |          |
| ndproxy | NDIS Proxy                              |                    |         |          |
| xy.sys  | c:\windows\system32\drivers\ndproxy.sys | Kernel Driver      | Yes     | Manual   |
|         |   | Running            | OK      | Normal   |
|         |   | No                 | Yes     |          |
| netbios | NetBIOS Interface                       |                    |         |          |
| os.sys  | c:\windows\system32\drivers\netbios.sys | File System Driver | Yes     |          |
|         |   | System             | Running | OK       |
|         |   | Normal             | No      | Yes      |
| netbt   | NetBios over Tcpip                      |                    |         |          |
| sys     | c:\windows\system32\drivers\netbt.sys   | Kernel Driver      | Yes     | System   |
|         |   | Running            | OK      | Normal   |
|         |   | No                 | Yes     |          |
| nfrd960 | nfrd960                                 |                    |         |          |
| 60.sys  | c:\windows\system32\drivers\nfrd960.sys | Kernel Driver      | No      | Disabled |
|         |   | Stopped            | OK      | Normal   |
|         |   | No                 | No      |          |
| npfs    | Npfs                                    |                    |         |          |
| ys      | c:\windows\system32\drivers\npfs.sys    | File System Driver | Yes     |          |
|         |   | System             | Running | OK       |
|         |   | Normal             | No      | Yes      |
| ntfs    | Ntfs                                    |                    |         |          |
| ys      | c:\windows\system32\drivers\ntfs.sys    | File System Driver | Yes     |          |
|         |   | Disabled           | Running | OK       |
|         |   | Normal             | No      | Yes      |
| null    | Null                                    |                    |         |          |
| ys      | c:\windows\system32\drivers\null.sys    | Kernel Driver      | Yes     | System   |
|         |   | Running            | OK      | Normal   |
|         |   | No                 | Yes     |          |
| partmgr | Partition Manager                       |                    |         |          |
| gr.sys  | c:\windows\system32\drivers\partmgr.sys | Kernel Driver      | Yes     | Boot     |
|         |   | Running            | OK      | Normal   |
|         |   | No                 | Yes     |          |
| pci     | PCI Bus Driver                          |                    |         |          |
| s       | c:\windows\system32\drivers\pci.sys     | Kernel Driver      | Yes     | Boot     |
|         |   | Running            | OK      | Critical |
|         |   | No                 | Yes     |          |
| pciide  | PCIIDE                                  |                    |         |          |
| .sys    | c:\windows\system32\drivers\pciide.sys  | Kernel Driver      | No      | Disabled |
|         |   | Stopped            | OK      | Normal   |
|         |   | No                 | No      |          |
| pcmcia  | Pcmcia                                  |                    |         |          |
| a.sys   | c:\windows\system32\drivers\pcmcia.sys  | Kernel Driver      | No      | Disabled |
|         |   | Stopped            | OK      | Normal   |
|         |   | No                 | No      |          |
| pdcomp  | PDCOMP                                  |                    |         |          |
| Driver  |   | Not Available      | Kernel  |          |
|         |   | No                 | Manual  | Stopped  |
|         |   | OK                 | Ignore  | No       |
|         |   | No                 |         |          |
| pdframe | PDFFRAME                                |                    |         |          |
| Driver  |   | Not Available      | Kernel  |          |
|         |   | No                 | Manual  | Stopped  |
|         |   | OK                 | Ignore  | No       |
|         |   | No                 |         |          |
| pdreli  | PDRELI                                  |                    |         |          |
| Driver  |   | Not Available      | Kernel  |          |
|         |   | No                 | Manual  | Stopped  |
|         |   | OK                 | Ignore  | No       |
|         |   | No                 |         |          |

|                            |   |               |        |          |
|----------------------------|---|---------------|--------|----------|
| pdframe                    | PDRFRAME                                  |               |        |          |
| Driver                     |   | No            | Manual | Stopped  |
|                            |   | OK            | Ignore | No       |
| pptpminiport               | WAN Miniport (PPTP)                       |               |        |          |
| tp.sys                     | c:\windows\system32\drivers\rasppptp.sys  | Kernel Driver | Yes    | Manual   |
|                            |   | Running       | OK     | Normal   |
|                            |   | No            | Yes    |          |
| processor                  | Processor Driver                          |               |        |          |
| ssr.sys                    | c:\windows\system32\drivers\processor.sys | Kernel Driver | Yes    | Manual   |
|                            |   | Running       | OK     | Normal   |
|                            |   | No            | Yes    |          |
| ptilink                    | Direct Parallel Link Driver               |               |        |          |
| .sys                       | c:\windows\system32\drivers\ptilink.sys   | Kernel Driver | Yes    | Manual   |
|                            |   | Running       | OK     | Normal   |
|                            |   | No            | Yes    |          |
| ql1080                     | ql1080                                    |               |        |          |
| 0.sys                      | c:\windows\system32\drivers\ql1080.sys    | Kernel Driver | No     | Disabled |
|                            |   | Stopped       | OK     | Normal   |
|                            |   | No            | No     |          |
| ql12160                    | ql12160                                   |               |        |          |
| 60.sys                     | c:\windows\system32\drivers\ql12160.sys   | Kernel Driver | No     | Disabled |
|                            |   | Stopped       | OK     | Normal   |
|                            |   | No            | No     |          |
| ql1240                     | ql1240                                    |               |        |          |
| 0.sys                      | c:\windows\system32\drivers\ql1240.sys    | Kernel Driver | No     | Disabled |
|                            |   | Stopped       | OK     | Normal   |
|                            |   | No            | No     |          |
| ql1280                     | ql1280                                    |               |        |          |
| 0.sys                      | c:\windows\system32\drivers\ql1280.sys    | Kernel Driver | No     | Disabled |
|                            |   | Stopped       | OK     | Normal   |
|                            |   | No            | No     |          |
| ql2100                     | ql2100                                    |               |        |          |
| 0.sys                      | c:\windows\system32\drivers\ql2100.sys    | Kernel Driver | No     | Disabled |
|                            |   | Stopped       | OK     | Normal   |
|                            |   | No            | No     |          |
| ql2200                     | ql2200                                    |               |        |          |
| 0.sys                      | c:\windows\system32\drivers\ql2200.sys    | Kernel Driver | No     | Disabled |
|                            |   | Stopped       | OK     | Normal   |
|                            |   | No            | No     |          |
| ql2300                     | QLLogic Fibre Channel STOR                |               |        |          |
| Miniport Driver (wia64 IP) |   |               |        |          |
| 0.sys                      | c:\windows\system32\drivers\ql2300.sys    | Kernel Driver | Yes    | Boot     |
|                            |   | Running       | OK     | Normal   |
|                            |   | No            | Yes    |          |
| qldirect                   | qldirect                                  |               |        |          |
| ct.sys                     | c:\windows\system32\drivers\qldirect.sys  | Kernel Driver | Yes    | Auto     |
|                            |   | Running       | OK     | Normal   |
|                            |   | No            | Yes    |          |
| rasacd                     | Remote Access Auto Connection             |               |        |          |
| Driver                     |   |               |        |          |
| d.sys                      | c:\windows\system32\drivers\rasacd.sys    | Kernel Driver | Yes    | System   |
|                            |   | Running       | OK     | Normal   |
|                            |   | No            | Yes    |          |
| rasl2tp                    | WAN Miniport (L2TP)                       |               |        |          |
| p.sys                      | c:\windows\system32\drivers\rasl2tp.sys   | Kernel Driver | Yes    | Manual   |
|                            |   | Running       | OK     | Normal   |
|                            |   | No            | Yes    |          |

|         |   |                    |          |          |
|---------|---|--------------------|----------|----------|
| rasppoe | Remote Access PPPOE Driver              |                    |          |          |
| poe.sys | c:\windows\system32\drivers\rasppoe.sys | Kernel Driver      | Yes      | Manual   |
|         |   | Running            | OK       | Normal   |
| raspti  | Direct Parallel                         |                    |          |          |
| .sys    | c:\windows\system32\drivers\raspti.sys  | Kernel Driver      | Yes      | Manual   |
|         |   | Running            | OK       | Normal   |
|         |   | No                 | Yes      |          |
| rdbss   | Rdbss                                   |                    |          |          |
| sys     | c:\windows\system32\drivers\rdbss.sys   | File System Driver | Yes      |          |
|         |   | System             | Running  | OK       |
|         |   | Normal             | No       | Yes      |
| rdpcdd  | RDPCCDD                                 |                    |          |          |
| d.sys   | c:\windows\system32\drivers\rdpcdd.sys  | Kernel Driver      | Yes      | System   |
|         |   | Running            | OK       | Ignore   |
|         |   | No                 | Yes      |          |
| rdpdr   | Terminal Server Device Redirector       |                    |          |          |
| Driver  |   |                    |          |          |
| sys     | c:\windows\system32\drivers\rdpdr.sys   | Kernel Driver      | Yes      | Manual   |
|         |   | Running            | OK       | Normal   |
|         |   | No                 | Yes      |          |
| rdpwd   | RDPWD                                   |                    |          |          |
| d.sys   | c:\windows\system32\drivers\rdpwd.sys   | Kernel Driver      | Yes      | Manual   |
|         |   | Running            | OK       | Ignore   |
|         |   | No                 | Yes      |          |
| redbook | Digital CD Audio Playback Filter        |                    |          |          |
| Driver  |   |                    |          |          |
| ok.sys  | c:\windows\system32\drivers\redbook.sys | Kernel Driver      | Yes      | System   |
|         |   | Running            | OK       | Normal   |
|         |   | No                 | Yes      |          |
| serenum | Serenum Filter Driver                   |                    |          |          |
| um.sys  | c:\windows\system32\drivers\serenum.sys | Kernel Driver      | Yes      | Manual   |
|         |   | Running            | OK       | Normal   |
|         |   | No                 | Yes      |          |
| serial  | Serial port driver                      |                    |          |          |
| sys     | c:\windows\system32\drivers\serial.sys  | Kernel Driver      | Yes      | System   |
|         |   | Running            | OK       | Ignore   |
|         |   | No                 | Yes      |          |
| sfloppy | Sfloppy                                 |                    |          |          |
| y.sys   | c:\windows\system32\drivers\sfloppy.sys | Kernel Driver      | No       | System   |
|         |   | Stopped            | OK       | Ignore   |
|         |   | No                 | No       |          |
| simbad  | Simbad                                  |                    |          |          |
| Driver  |   | Not Available      | Kernel   | Stopped  |
|         |   | No                 | Disabled | Stopped  |
|         |   | OK                 | Normal   | No       |
|         |   | No                 |          |          |
| srv     | Srv                                     |                    |          |          |
| s       | c:\windows\system32\drivers\server.sys  | File System Driver | Yes      |          |
|         |   | Manual             | Running  | OK       |
|         |   | Normal             | No       | Yes      |
| swenum  | Software Bus Driver                     |                    |          |          |
| um.sys  | c:\windows\system32\drivers\swenum.sys  | Kernel Driver      | Yes      | Manual   |
|         |   | Running            | OK       | Normal   |
|         |   | No                 | Yes      |          |
| symc8xx | symc8xx                                 |                    |          |          |
| 8xx.sys | c:\windows\system32\drivers\symc8xx.sys | Kernel Driver      | No       | Disabled |
|         |   | Stopped            | OK       | Normal   |
|         |   | No                 | No       |          |

```

symmpi symmpi
c:\windows\system32\drivers\symm
pi.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 No Disabled
Stopped OK Normal
No No
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\tdpip
e.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\termd
d.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.s
ys File System Driver No
Disabled Stopped OK
Normal No No
usbccgp Microsoft USB Generic Parent Driver
c:\windows\system32\drivers\usbcc
gp.sys Kernel Driver Yes Manual
Running OK Normal
No Yes
usbhci Microsoft USB 2.0 Enhanced Host
Controller Miniport Driver
c:\windows\system32\drivers\usbeh
ci.sys Kernel Driver Yes Manual
Running OK Normal
No Yes
usbhub USB2 Enabled Hub
c:\windows\system32\drivers\usbhu
b.sys Kernel Driver Yes Manual
Running OK Normal
No Yes
usbohci Microsoft USB Open Host Controller
Miniport Driver
c:\windows\system32\drivers\usboh
ci.sys Kernel Driver Yes Manual
Running OK Normal
No Yes
usbstor USB Mass Storage Driver
c:\windows\system32\drivers\usbst
or.sys Kernel Driver Yes Manual
Running OK Normal
No Yes

```

```

vgasave VGA Display Controller.
c:\windows\system32\drivers\vga.s
ys Kernel Driver Yes System
Running OK Ignore
viaide viaide
c:\windows\system32\drivers\viaide
.sys Kernel Driver No Disabled
Stopped OK Normal
No No
volsnap Storage volumes
c:\windows\system32\drivers\volsn
ap.sys Kernel Driver Yes Boot
Running OK Normal
No Yes
wanarp Remote Access IP ARP Driver
c:\windows\system32\drivers\wana
rp.sys Kernel Driver Yes Manual
Running OK Normal
No Yes
wdica WDICA Not Available Kernel
Driver No Manual Stopped
OK Ignore 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
Microsoft System Management BIOS Driver
Yes SYSTEM
5.2.3790.1830
10/1/2002 (Standard system
devices) machine.inf Not Available
ROOT\SYSTEM\0001
Plug and Play Software Device Enumerator
Yes SYSTEM
5.2.3790.1830
10/1/2002 (Standard system
devices) machine.inf Not Available
ROOT\SYSTEM\0000
Terminal Server Mouse Driver Yes
SYSTEM 5.2.3790.1830
10/1/2002 (Standard system
devices) machine.inf Not Available
ROOT\RDPMOU\0000
Terminal Server Keyboard Driver Yes
SYSTEM 5.2.3790.1830
10/1/2002 (Standard system
devices) machine.inf Not Available
ROOT\RDPKBD\0000
Terminal Server Device Redirector Yes
SYSTEM 5.2.3790.1830
10/1/2002 (Standard system
devices) machine.inf Not Available
ROOT\RDPPDR\0000
Direct Parallel Yes NET
5.2.3790.1830
10/1/2002 Microsoft
netrasa.inf Not Available
ROOT\MS_PTMINIIMPORT\0000

WAN Miniport (PPTP) Yes NET
5.2.3790.1830
10/1/2002 Microsoft
netrasa.inf Not Available
ROOT\MS_PPTPMINIIMPORT\0000

```

```

WAN Miniport (PPPOE) Yes NET
5.2.3790.1830
10/1/2002 Microsoft
netrasa.inf Not Available
ROOT\MS_PPPOEMINIIMPORT\0000
WAN Miniport (IP) Yes NET
5.2.3790.1830
10/1/2002 Microsoft
netrasa.inf Not Available
ROOT\MS_NDISWANIP\0000
WAN Miniport (L2TP) Yes NET
5.2.3790.1830
10/1/2002 Microsoft
netrasa.inf Not Available
ROOT\MS_L2TPMINIIMPORT\0000
Video CodecsYes MEDIA
5.2.3790.0 10/1/2002
(Standard system devices) wave.inf
Not Available
ROOT\MEDIA\MS_MMVID
Legacy Video Capture Devices Yes
MEDIA 5.2.3790.0
10/1/2002 (Standard system
devices) wave.inf Not Available
ROOT\MEDIA\MS_MMVCD
Media Control Devices Yes MEDIA
5.2.3790.0 10/1/2002
(Standard system devices) wave.inf
Not Available
ROOT\MEDIA\MS_MMMCI
Legacy Audio Drivers Yes MEDIA
5.2.3790.0 10/1/2002
(Standard system devices) wave.inf
Not Available
ROOT\MEDIA\MS_MMDRV
Audio CodecsYes MEDIA
5.2.3790.0 10/1/2002
(Standard system devices) wave.inf
Not Available
ROOT\MEDIA\MS_MMACM
Remote Access IP ARP Driver Not
Available LEGACYDRIVER Not
Available Not Available Not
Available Not Available
ROOT\LEGACY_WANARP\0000
volsnap Not Available LEGACYDRIVER
Available Not Available Not Available
ROOT\LEGACY_VOLSNAP\0000
VGA Display Controller. Not Available
LEGACYDRIVER Not
Available Not Available Not Available Not
Available Not Available
ROOT\LEGACY_VGASAVE\0000
TDTCP Not Available LEGACYDRIVER
Available Not Available Not Available Not
Available Not Available
ROOT\LEGACY_TDTCP\0000
TCP/IP Protocol Driver Not Available
LEGACYDRIVER Not
Available Not Available Not Available Not
Available Not Available
ROOT\LEGACY_TCPIP\0000

```

sacdrv Not Available LEGACYDRIVER  
 Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_SACDRV\0000

RDPWD Not Available LEGACYDRIVER  
 Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_RDPWD\0000

RDPCCD Not Available LEGACYDRIVER  
 Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_RDPCCD\0000

Remote Access Auto Connection Driver Not Available  
 LEGACYDRIVER Not Available  
 Available Not Available Not Available Not Available  
 Available Not Available  
 ROOT\LEGACY\_RASACD\0000

qldirect Not Available LEGACYDRIVER  
 Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_QLDIRECT\0000

Partition Manager Not Available  
 LEGACYDRIVER Not Available  
 Available Not Available Not Available Not Available  
 Available Not Available  
 ROOT\LEGACY\_PARTMGR\0000

Null Not Available LEGACYDRIVER  
 Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_NULL\0000

NetBios over Tcpip Not Available  
 LEGACYDRIVER Not Available  
 Available Not Available Not Available Not Available  
 Available Not Available  
 ROOT\LEGACY\_NETBT\0000

NDProxy Not Available LEGACYDRIVER  
 Not Available Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_NDPROXY\0000

NDIS Usermode I/O Protocol Not Available  
 Available LEGACYDRIVER Not Available  
 Available Not Available Not Available Not Available  
 Available Not Available  
 ROOT\LEGACY\_NDISUIO\0000

Remote Access NDIS TAPI Driver Not Available  
 Available LEGACYDRIVER Not Available  
 Available Not Available Not Available Not Available  
 Available Not Available  
 ROOT\LEGACY\_NDISTAPI\0000

NDIS System Driver Not Available  
 LEGACYDRIVER Not Available  
 Available Not Available Not Available Not Available  
 Available Not Available  
 ROOT\LEGACY\_NDIS\0000

mountmgr Not Available LEGACYDRIVER  
 Not Available Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_MOUNTMGR\0000

lpxnds Not Available LEGACYDRIVER  
 Not Available Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_LPXNDS\0000

lpxftr Not Available LEGACYDRIVER  
 Not Available Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_LPXFTR\0000

ksecdd Not Available LEGACYDRIVER  
 Not Available Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_KSECDD\0000

IPSEC driver Not Available LEGACYDRIVER  
 Not Available Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_IPSEC\0000

IP Network Address Translator Not Available  
 Available LEGACYDRIVER Not Available  
 Available Not Available Not Available Not Available  
 Available Not Available  
 ROOT\LEGACY\_IPNAT\0000

hpciss Not Available LEGACYDRIVER  
 Not Available Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_HPCISS\0000

Generic Packet Classifier Not Available  
 LEGACYDRIVER Not Available  
 Available Not Available Not Available Not Available  
 Available Not Available  
 ROOT\LEGACY\_GPC\0000

Fips Not Available LEGACYDRIVER  
 Not Available Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_FIPS\0000

dmload Not Available LEGACYDRIVER  
 Not Available Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_DMLOAD\0000

dmboot Not Available LEGACYDRIVER  
 Not Available Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_DMBOOT\0000

CRC Disk Filter Driver Not Available  
 LEGACYDRIVER Not Available  
 Available Not Available Not Available Not Available  
 Available Not Available  
 ROOT\LEGACY\_CRCDISK\0000

Beep Not Available LEGACYDRIVER  
 Not Available Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_BEEP\0000

AFD Not Available LEGACYDRIVER  
 Not Available Not Available Not Available Not Available  
 Available Not Available Not Available  
 ROOT\LEGACY\_AFD\0000

Generic volume Yes VOLUME  
 5.2.3790.1830  
 10/1/2002 Microsoft  
 volume.inf Not Available  
 STORAGE\VOLUME\1&30A96598&0  
 &GPTPARTITION{75BDA503-0078-4E05-BA40-CAE3B344E79B}

Generic volume Yes VOLUME  
 5.2.3790.1830  
 10/1/2002 Microsoft  
 volume.inf Not Available  
 STORAGE\VOLUME\1&30A96598&0

Generic volume Yes VOLUME  
 5.2.3790.1830  
 10/1/2002 Microsoft  
 volume.inf Not Available  
 STORAGE\VOLUME\1&30A96598&0  
 &GPTPARTITION{2310DE8B-8CFA-4601-A523-268E7D47A018}

Generic volume Yes VOLUME  
 5.2.3790.1830  
 10/1/2002 Microsoft  
 volume.inf Not Available  
 STORAGE\VOLUME\1&30A96598&0  
 &GPTPARTITION{3A79A7E1-4BCB-49C3-93D9-94F291C555CC}

Generic volume Yes VOLUME  
 5.2.3790.1830  
 10/1/2002 Microsoft  
 volume.inf Not Available  
 STORAGE\VOLUME\1&30A96598&0  
 &GPTPARTITION{B1645F95-3FFE-4E50-A3CB-6A82A0BCCEFB}

Generic volume Yes VOLUME  
 5.2.3790.1830  
 10/1/2002 Microsoft  
 volume.inf Not Available  
 STORAGE\VOLUME\1&30A96598&0  
 &GPTPARTITION{A8608377-7672-401D-81B7-9B0EF269DAD9}

Generic volume Yes VOLUME  
 5.2.3790.1830  
 10/1/2002 Microsoft  
 volume.inf Not Available  
 STORAGE\VOLUME\1&30A96598&0  
 &GPTPARTITION{68BD04B9-7A8A-4790-B8B9-4B132299519C}

Generic volume Yes VOLUME  
 5.2.3790.1830  
 10/1/2002 Microsoft  
 volume.inf Not Available  
 STORAGE\VOLUME\1&30A96598&0  
 &GPTPARTITION{D7759FBF-CFEC-4799-8B32-C6680BCAFA9F}

Generic volume Yes VOLUME  
 5.2.3790.1830  
 10/1/2002 Microsoft  
 volume.inf Not Available  
 STORAGE\VOLUME\1&30A96598&0  
 &GPTPARTITION{36E5A569-8B42-4889-95C5-57F6C6CAB3AF}

Generic volume Yes VOLUME  
 5.2.3790.1830  
 10/1/2002 Microsoft  
 volume.inf Not Available  
 STORAGE\VOLUME\1&30A96598&0  
 &GPTPARTITION{A913D5A0-778B-428F-93C1-A68BEC045C9B}

Generic volume Yes VOLUME  
 5.2.3790.1830  
 10/1/2002 Microsoft  
 volume.inf Not Available  
 STORAGE\VOLUME\1&30A96598&0  
 &GPTPARTITION{8D60B71F-9CE2-4975-98A1-C71895357611}

Generic volume Yes VOLUME  
 5.2.3790.1830  
 10/1/2002 Microsoft  
 volume.inf Not Available  
 STORAGE\VOLUME\1&30A96598&0  
 &GPTPARTITION{1C64E3CC-0A39-437F-9470-A4071E48E60D}



Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{202D8430-564B-45C4-B5E3-702C028B6839}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{C305BF6A-CE81-43E6-A29C-DC3944527568}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{DD83BE58-C36B-4DA2-991B-023C286E5EE4}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{0DF4836E-E493-4797-9BF6-DCEE52BA2406}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{D4021FFF-7168-4E59-BB62-CAD9A6435732}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{1060C9F2-3A83-4F92-9A43-DA9DFDD1DD0B}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{3788C7CD-364F-4869-B1E4-9888F3BD9EF3}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{34242EBE-E08B-4378-9739-611A479DD28C}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{6B9D5ACC-84F1-4505-A236-213E9259CCCA}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{516290FF-940B-4580-B4E1-774AE8036824}

Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{8CBFD46D-83B2-4933-979F-8ED3387409C8}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{59C4D8EE-0D93-4EE9-804A-BE13454F7DB0}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{E015D645-BDA2-47D7-BA48-22E730591117}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{97AF8CFA-8B7A-49D4-A428-96A1A75D0EC6}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{5DDB54C0-1564-4DD3-8FCF-B7D182046C6E}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{080EE576-D86D-4B83-85D0-3D4AA3C4F6C6}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{15BE146E-7C93-4A79-AF6B-0DD6B10A31DD}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{A30B816E-9634-490C-933E-F4DCE89BD611}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{50E68DB7-76A2-4A06-8944-2F66DA99C9A2}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{4898370B-F737-4446-8996-619C87BE3A06}

Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{3C1EDB04-6D36-4A9D-B926-024D3DD18A81}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{2F3FE498-F292-4D9A-9C47-25E57AA50F7F}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{E8DDAB7D-0E72-4E70-BE02-A8BDOC79D2D8}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{14B0856E-F587-4C88-9A0A-8DD7E9DBE7EB}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{023033A1-E8ED-4551-A38B-FC71EF725846}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{3E16D29E-A4D0-4B92-B3D1-626B3866B655}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{9075F93F-F8C3-47E2-9D7B-EBB57204F840}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{E35116EE-B45C-4448-AF72-A460EC0D8587}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{474868E1-E69E-4725-9BBC-CF832001828C}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{31AD0111-0026-441F-8160-25D78C6368C7}

Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{32E03D01-CCA6-462E-ACE7-  
E9A4732E09C1}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{B82C53A3-0AEE-4F67-9F44-  
7447A7052F36}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{EEED8318-0CAE-48BC-A4D1-  
6A8AFCFD7732}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{082CCFA4-03EF-4A1D-94B2-  
6DA16FEFB03F}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{BA06E4AA-7C19-484C-A6AE-  
933F7F3ECC23}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{6B20823B-2F3F-4AC2-AAF1-  
704130ABFA4C}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{4BD380C2-B60B-48E8-8473-  
0F31BCE2A2A8}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{5F73B139-CFA2-4A67-BDB3-  
884CCF13C11F}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{D5328481-E0BC-4619-A6BB-  
2E23EFD66376}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{C9B67919-DB27-4261-BBA2-  
C34ACA052C76}

Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{0D28C149-CF6A-43B2-AAF2-  
B67510958A2B}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{181EAEF9-3BAA-4E06-A970-  
9767BE553286}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{B3519DD8-AD01-44F9-B3B4-  
24815F5E7701}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{6874CEA4-BDEB-4CD2-AD04-  
C07AA6DC3E72}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{EBA9F8F2-3740-4D07-B217-  
9536FEA42FD5}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{0F52414C-F919-4FCD-AF26-  
A7D06B609ED8}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{2A34AC12-160D-4347-9D30-  
ADA5BE163C34}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{B6DAFD81-AAAB-4B9E-8B22-  
B24239C67597}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{13BF6E96-C746-4DE4-A3A5-  
2091AF9F5D05}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{1E01A2B4-2968-4F64-99F7-  
EA172575ADCC}

Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{34852B3F-9094-4A5F-9AA7-  
FEA8805F6EE3}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{D9818F1B-DD6F-4928-9C81-  
F0330971DADF}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{1062AB55-D569-43D9-B647-  
9886190F53CC}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{10E0FAA1-2142-4F1B-99A5-  
B9AB6182E7FA}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{AEA56DE1-CACC-498A-9CF9-  
A1909A80B6D8}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{AA8CDDFC-A456-4D3B-A29C-  
3405F8BCA18B}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{0B7651D5-B0E9-41BE-902D-  
7CE373DEC8A6}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{A0BA8248-460D-43E4-8344-  
EDE5BBD26E40}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{633698DC-D9F9-4472-95F7-  
6EDD129BA6A0}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\OLUME\1&30A96598&0  
&GPTPARTITION{5404C031-2301-46D0-8E8D-  
5CC498130325}

Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{0F81F0D-990B-4719-99E2-5DFB136E42A1}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{63357D9D-9069-4DB7-A81A-E37D2C56ADC7}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{0F760452-053D-4459-BD20-B0122FCBA760}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{F17D1F92-146C-426F-84B1-5CBB1C834009}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{7123E534-11C1-4E9C-A97F-07FB294046FF}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{5284C31C-03D2-11DB-975B-000000000000}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{515379FC-03D2-11DB-975B-000000000000}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{50BAF1F0-03D2-11DB-975B-000000000000}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{50BAEE6C-03D2-11DB-975B-000000000000}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{92A00974-7C5D-4003-A17E-1D4B99185286}

Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{7C0A38A8-C2E0-4712-958F-BE50E0D0A43}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{A23D714E-AA86-42D4-9815-DC42C5202491}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{B79E3AB7-2E52-45B8-BA2B-441FA6365253}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{F2126C96-5A05-41C7-BAF1-D1A0B357DC13}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{76B0B633-8B0B-4F5B-A564-D9679D708520}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{AF3C3BFC-F1EF-4944-9A6D-C0B6AFE95A1E}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{EC44741F-92C1-455F-890F-196A38654A96}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{DD8234B6-93E3-4DCF-B694-F2B1ED7DD63C}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{A40A3DCB-6C29-42DC-A0AC-2B758C8F094A}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{07810AF1-4289-4CC1-8488-7A15875E0D56}

Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{3087436F-1DE8-49F6-923B-1A861498FE48}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{ECE6731F-5A08-402C-9E65-5FE0061E7E2B}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{57BE54AA-E880-4F74-9F22-0B6E07E8A57D}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{0C759C0F-8717-4ADD-A276-28D42554EDFF}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{FE436629-1E35-4CA3-902E-CAE8E8ADA22D}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{91EF35EA-9335-439E-BA0B-78F28C16105A}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{D880440E-8EA2-4984-B9AB-366AD78841AA}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{66C23836-A587-4FBB-8F05-CC9298135C8A}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{1AFA6B77-CD6E-4FAD-A4B5-3402AC77646A}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{F1BE97A3-1A79-40FD-BD94-201FAB604EEF}

Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{53F91992-E4B5-4046-B265-A66785FBF77A}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{06CB62D-0DE6-4E65-A3A1-F041E53B8E11}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{07759F13-58A4-4938-BF9F-FE29F00E02C7}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{E6453BF9-0835-47D3-87D3-00B018800330}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{FFE02990-65A4-4907-A339-E847EE29A4FD}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{1E3425C2-02FC-48E1-96CD-4774A4B892A8}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{92C7CE93-3BD0-4B85-8475-9FAFEE4F544C}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{040AFE95-1A20-4A72-BEF4-6545629A2B59}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{0ADCEA85-763A-42A8-9969-CD38028E4F94}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{D286D775-155D-466F-9979-AD4A7F262E8D}

Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{3E8D8B11-ACEA-4348-B39A-FC75B2CB85E8}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{465A97E0-A449-4116-A672-08B19371BC71}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{18EF676B-13EF-4179-AF4C-1288ACCE0E302}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{4AA84A49-C5E0-4E10-A54E-2F6E4F030C77}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{F3AE6D3E-5341-40B5-89DF-E6BEB3523D26}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{6990BFA2-6A9F-4377-ABAF-E2FB1337C096}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{BE9F9D3E-DAF8-4EE7-8158-401DFD4A263F}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{E2CC1585-B357-4850-887A-2E60691085F6}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{ABB27B9-FD1E-404E-B69B-C65CF867E510}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{7B339AA7-62BE-4572-A957-A0809F84235F}

Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{7408B350-71AD-4E57-A8FF-2036D08A6E3A}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{32425D34-B17A-46AA-A8F5-88580356EB4A}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{F14DE056-205B-49B2-9E56-F700CF743340}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{A4E67D11-2FE6-4B9F-9EDC-30567E96D0AC}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{502F4D82-6B4D-48F0-9EA1-8A6C6C76812A}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{783F4C2E-D5AD-47E6-8618-104A428FEA05}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{CFD68A99-B865-4405-A2A1-4F3B021C627E}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{2744D093-F97E-4E82-BF55-238C32884D4B}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{0338E0CE-5CA2-4142-99B2-54F8151989F3}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{1AACF7F4-155B-43DC-BDEF-F24DD1310E20}

Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{A387B0A8-D528-4D53-8B18-51A482592535}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{E886D2DB-0DB0-424B-AFB7-8CF7B89ACA13}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{2AAF12B6-B027-4ACD-B3CF-D7B8E7715FE2}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{AA6CC4C-3DAD-4CE7-A783-412CE4D7DCB4}  
Generic volume Yes VOLUME  
5.2.3790.1830  
10/1/2002 Microsoft  
volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0  
&GPTPARTITION{628243D3-4D37-4B63-9A6A-7342BDEE5352}  
Volume Manager Yes SYSTEM  
5.2.3790.1830  
10/1/2002 (Standard system  
devices) machine.inf Not Available  
ROOT\FTDISK\0000  
Logical Disk Manager Yes SYSTEM  
5.2.3790.1830  
10/1/2002 (Standard system  
devices) machine.inf Not Available  
ROOT\DMIO\0000  
Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_15  
Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_14  
Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_13  
Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_12  
Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_11

Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_10  
Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_9  
Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_8  
Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_7  
Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_6  
Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_5  
Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_4  
Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_3  
Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_2  
Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_1  
Processor Yes PROCESSOR  
5.2.3790.1830  
10/1/2002 (Standard processor  
types) cpu.inf Not Available  
ACPI\GENUINEINTEL\_-\_IA64\_FAMILY\_32\_MODEL\_0\_0  
ACPI Fixed Feature Button Yes SYSTEM  
5.2.3790.1830  
10/1/2002 (Standard system  
devices) machine.inf Not Available  
ACPI\FIXEDBUTTON\2&DABA3FF&0

Qlogic processor device Yes SYSTEM  
5.2.3790.1830  
10/1/2002 QLOGIC  
scsdev.inf Not Available  
SCSI\PROCESSOR&VEN\_QLOGIC&P  
ROD\_PSEUDO\_DEVICE&REV\_5&2A49C7E1&0&0  
7F0  
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\_4.48\5&2A49C7E1&0&0  
12  
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\_4.48\5&2A49C7E1&0&0  
11  
StorageWorks MSA1000 Yes SYSTEM  
5.32.0.64 9/9/2005 Hewlett-  
Packard Company oem16.inf Not  
Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD\_  
\_MSA1000&REV\_4.48\5&2A49C7E1&0&010  
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\_4.48\5&2A49C7E1&0&0  
02  
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\_4.48\5&2A49C7E1&0&0  
01  
StorageWorks MSA1000 Yes SYSTEM  
5.32.0.64 9/9/2005 Hewlett-  
Packard Company oem16.inf Not  
Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD\_  
\_MSA1000&REV\_4.48\5&2A49C7E1&0&000  
QLogic Fibre Channel Adapter Yes  
SCSIADAPTER 9.1.0.18  
10/11/2005 QLogic oem7.inf  
Not Available  
PCI\VEN\_1077&DEV\_2312&SUBSYS\_01010E11&REV\_02\4&1E72F330&0&09  
Qlogic processor device Yes SYSTEM  
5.2.3790.1830  
10/1/2002 QLOGIC  
scsdev.inf Not Available  
SCSI\PROCESSOR&VEN\_QLOGIC&P  
ROD\_PSEUDO\_DEVICE&REV\_5&1790173C&0&0  
7F0  
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\_4.48\5&1790173C&0&0  
02

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\_4.48\581790173C&0&0  
01  
StorageWorks MSA1000 Yes SYSTEM  
5.32.0.64 9/9/2005 Hewlett-  
Packard Company oem16.inf Not  
Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD\_  
\_MSA1000&REV\_4.48\581790173C&0&000

QLogic Fibre Channel Adapter Yes  
SCSIADAPTER 9.1.0.18  
10/11/2005 QLogic oem7.inf  
Not Available  
PCI\VEN\_1077&DEV\_2312&SUBSYS  
\_01010E11&REV\_02\4&1E72F330&0&08

PCI bus Yes SYSTEM  
5.2.3790.1830  
10/1/2002 (Standard system  
machine.inf Not Available  
ACPI\HWP0002\700

Qlogic processor device Yes SYSTEM  
5.2.3790.1830  
10/1/2002 QLOGIC  
scsidev.inf Not Available  
SCSI\PROCESSOR&VEN\_QLOGIC&P  
ROD\_PSEUDO\_DEVICE&REV\_5&2FF938CE&0&0  
7F0

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\_4.48\582FF938CE&0&0  
12

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\_4.48\582FF938CE&0&0  
11

StorageWorks MSA1000 Yes SYSTEM  
5.32.0.64 9/9/2005 Hewlett-  
Packard Company oem16.inf Not  
Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD\_  
\_MSA1000&REV\_4.48\582FF938CE&0&010

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\_4.48\582FF938CE&0&0  
02

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\_4.48\582FF938CE&0&0  
01

StorageWorks MSA1000 Yes SYSTEM  
5.32.0.64 9/9/2005 Hewlett-  
Packard Company oem16.inf Not  
Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD\_  
\_MSA1000&REV\_4.48\582FF938CE&0&000

QLogic Fibre Channel Adapter Yes  
SCSIADAPTER 9.1.0.18  
10/11/2005 QLogic oem7.inf  
Not Available  
PCI\VEN\_1077&DEV\_2312&SUBSYS  
\_01010E11&REV\_02\4&5D9C86&0&08

Qlogic processor device Yes SYSTEM  
5.2.3790.1830  
10/1/2002 QLOGIC  
scsidev.inf Not Available  
SCSI\PROCESSOR&VEN\_QLOGIC&P  
ROD\_PSEUDO\_DEVICE&REV\_5&7181F6C&0&07  
F0

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\_4.48\587181F6C&0&01  
2

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\_4.48\587181F6C&0&01  
1

StorageWorks MSA1000 Yes SYSTEM  
5.32.0.64 9/9/2005 Hewlett-  
Packard Company oem16.inf Not  
Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD\_  
\_MSA1000&REV\_4.48\587181F6C&0&010

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\_4.48\587181F6C&0&00  
2

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\_4.48\587181F6C&0&00  
1

StorageWorks MSA1000 Yes SYSTEM  
5.32.0.64 9/9/2005 Hewlett-  
Packard Company oem16.inf Not  
Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD\_  
\_MSA1000&REV\_4.48\587181F6C&0&000

QLogic Fibre Channel Adapter Yes  
SCSIADAPTER 9.1.0.18  
10/11/2005 QLogic oem7.inf  
Not Available  
PCI\VEN\_1077&DEV\_2312&SUBSYS  
\_01010E11&REV\_02\4&5D9C86&0&08

PCI bus Yes SYSTEM  
5.2.3790.1830  
10/1/2002 (Standard system  
machine.inf Not Available  
ACPI\HWP0002\600

Qlogic processor device Yes SYSTEM  
5.2.3790.1830  
10/1/2002 QLOGIC  
scsidev.inf Not Available  
SCSI\PROCESSOR&VEN\_QLOGIC&P  
ROD\_PSEUDO\_DEVICE&REV\_5&9A56C22&0&07  
F0

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\_4.48\589A56C22&0&01  
1

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\_4.48\589A56C22&0&01  
1

StorageWorks MSA1000 Yes SYSTEM  
5.32.0.64 9/9/2005 Hewlett-  
Packard Company oem16.inf Not  
Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD\_  
\_MSA1000&REV\_4.48\589A56C22&0&010

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\_4.48\589A56C22&0&00  
2

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\_4.48\589A56C22&0&00  
1

StorageWorks MSA1000 Yes SYSTEM  
5.32.0.64 9/9/2005 Hewlett-  
Packard Company oem16.inf Not  
Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD\_  
\_MSA1000&REV\_4.48\589A56C22&0&000

QLogic Fibre Channel Adapter Yes  
SCSIADAPTER 9.1.0.18  
10/11/2005 QLogic oem7.inf  
Not Available  
PCI\VEN\_1077&DEV\_2312&SUBSYS  
\_01010E11&REV\_02\4&24543408&0&09

Qlogic processor device Yes SYSTEM  
5.2.3790.1830  
10/1/2002 QLOGIC  
scsidev.inf Not Available  
SCSI\PROCESSOR&VEN\_QLOGIC&P  
ROD\_PSEUDO\_DEVICE&REV\_5&1C5F1CC7&0&0  
7F0

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\_4.48\581C5F1CC7&0&0  
12

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\_4.48\581C5F1CC7&0&0  
11

StorageWorks MSA1000 Yes SYSTEM  
5.32.0.64 9/9/2005 Hewlett-  
Packard Company oem16.inf Not  
Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD  
\_MSA1000&REV\_4.48\5&1C5F1CC7&0&010

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\_4.48\5&1C5F1CC7&0&0  
02

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\_4.48\5&1C5F1CC7&0&0  
01

StorageWorks MSA1000 Yes SYSTEM  
5.32.0.64 9/9/2005 Hewlett-  
Packard Company oem16.inf Not  
Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD  
\_MSA1000&REV\_4.48\5&1C5F1CC7&0&000

QLogic Fibre Channel Adapter Yes  
SCSIADAPTER 9.1.0.18  
10/11/2005 QLogic oem7.inf  
Not Available  
PCI\VEN\_1077&DEV\_2312&SUBSYS  
\_01010E11&REV\_02\4&24543408&0&08

PCI bus Yes SYSTEM  
5.2.3790.1830  
10/1/2002 (Standard system  
machine.inf Not Available  
ACPI\HWP0002\500

Intel(R) PRO/1000 MT Dual Port Server Adapter  
Yes NET 8.5.14.0  
3/8/2005 Intel  
oem10.inf Not Available  
PCI\VEN\_8086&DEV\_1079&SUBSYS  
\_12A6103C&REV\_03\4&15291AB&0&11

Intel(R) PRO/1000 MT Dual Port Server Adapter  
Yes NET 8.5.14.0  
3/8/2005 Intel  
oem10.inf Not Available  
PCI\VEN\_8086&DEV\_1079&SUBSYS  
\_12A6103C&REV\_03\4&15291AB&0&10

Disk drive Yes DISKDRIVE  
5.2.3790.0 10/1/2002  
(Standard disk drives) disk.inf  
Not Available  
SCSI\DISK&VEN\_COMPAQ&PROD\_L  
OGICAL\_VOLUME&REV\_1.20\5&2D46D291&0&00  
0400

HP Virtual LUN Yes SYSTEM  
5.2.3790.1830  
10/1/2002 Compaq  
scsidev.inf Not Available  
SCSI\OTHER&VEN\_COMPAQ&PROD  
\_SCSI\_COMMUNICATE&REV\_CIS2\5&2D46D291  
&0&000000

Smart Array P600 Controller Yes  
SCSIADAPTER 5.6.0.64  
8/29/2005 Hewlett-Packard  
Company oem9.inf Not Available  
PCI\VEN\_103C&DEV\_3220&SUBSYS  
\_3225103C&REV\_00\4&15291AB&0&08

PCI bus Yes SYSTEM  
5.2.3790.1830  
10/1/2002 (Standard system  
machine.inf Not Available  
ACPI\HWP0002\500

QLogic processor device Yes SYSTEM  
5.2.3790.1830  
10/1/2002 QLOGIC  
scsidev.inf Not Available  
SCSI\PROCESSOR&VEN\_QLOGIC&P  
ROD\_PSEUDO\_DEVICE&REV\_5&197AF1BB&0&0  
7F0

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\_4.48\5&197AF1BB&0&0  
02

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\_4.48\5&197AF1BB&0&0  
01

StorageWorks MSA1000 Yes SYSTEM  
5.32.0.64 9/9/2005 Hewlett-  
Packard Company oem16.inf Not  
Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD  
\_MSA1000&REV\_4.48\5&197AF1BB&0&000

QLogic Fibre Channel Adapter Yes  
SCSIADAPTER 9.1.0.18  
10/11/2005 QLogic oem7.inf  
Not Available  
PCI\VEN\_1077&DEV\_2312&SUBSYS  
\_01010E11&REV\_02\4&19EBB955&0&09

QLogic processor device Yes SYSTEM  
5.2.3790.1830  
10/1/2002 QLOGIC  
scsidev.inf Not Available  
SCSI\PROCESSOR&VEN\_QLOGIC&P  
ROD\_PSEUDO\_DEVICE&REV\_5&6C14116&0&07  
F0

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\_4.48\5&6C14116&0&0&01  
2

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\_4.48\5&6C14116&0&0&01  
1

StorageWorks MSA1000 Yes SYSTEM  
5.32.0.64 9/9/2005 Hewlett-  
Packard Company oem16.inf Not  
Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD  
\_MSA1000&REV\_4.48\5&6C14116&0&010

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\_4.48\5&6C14116&0&0&00  
0

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\_4.48\5&6C14116&0&0&00  
01

StorageWorks MSA1000 Yes SYSTEM  
5.32.0.64 9/9/2005 Hewlett-  
Packard Company oem16.inf Not  
Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD  
\_MSA1000&REV\_4.48\5&6C14116&0&0000

QLogic Fibre Channel Adapter Yes  
SCSIADAPTER 9.1.0.18  
10/11/2005 QLogic oem7.inf  
Not Available  
PCI\VEN\_1077&DEV\_2312&SUBSYS  
\_01010E11&REV\_02\4&19EBB955&0&08

PCI bus Yes SYSTEM  
5.2.3790.1830  
10/1/2002 (Standard system  
machine.inf Not Available  
ACPI\HWP0002\300

QLogic processor device Yes SYSTEM  
5.2.3790.1830  
10/1/2002 QLOGIC  
scsidev.inf Not Available  
SCSI\PROCESSOR&VEN\_QLOGIC&P  
ROD\_PSEUDO\_DEVICE&REV\_5&24BC464F&0&0  
7F0

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\_4.48\5&24BC464F&0&0&0  
12

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\_4.48\5&24BC464F&0&0&0  
11

StorageWorks MSA1000 Yes SYSTEM  
5.32.0.64 9/9/2005 Hewlett-  
Packard Company oem16.inf Not  
Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD  
\_MSA1000&REV\_4.48\5&24BC464F&0&010

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\_4.48\5&24BC464F&0&0&0  
02





```

USB Composite Device Yes USB
5.2.3790.1830
10/1/2002 (Standard USB Host
Controller) usb.inf Not Available
USB\VID_03F0&PID_1126\6&3B37
DE0D&0&1
USB Root Hub Yes USB
5.2.3790.1830
10/1/2002 (Standard USB Host
Controller) usbport.inf Not Available
USB\ROOT_HUB\5&L1786A2A0&0

NEC PCI to USB Open Host Controller Yes
USB 5.2.3790.1830
10/1/2002 NEC
usbport.inf Not Available
PCI\VEN_1033&DEV_0035&SUBSYS
_022603F0&REV_43\4&4F5EBC7&0&10
HP Management Processor Yes SYSTEM
5.0.3790.1 7/28/2005 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_103C&DEV_1048&SUBSYS
_1301103C&REV_00\4&4F5EBC7&0&0A
HP Management Shared Memory Device Yes
SYSTEM 5.0.3790.1
7/27/2005 Hewlett-Packard
oem8.inf Not Available
PCI\VEN_103C&DEV_1302&SUBSYS
_1302103C&REV_00\4&4F5EBC7&0&09
HP Management Processor Yes SYSTEM
5.0.3790.1 7/28/2005 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_103C&DEV_1303&SUBSYS
_1303103C&REV_00\4&4F5EBC7&0&08
PCI bus Yes SYSTEM
5.2.3790.1830
10/1/2002 (Standard system
devices) machine.inf Not Available
ACPI\HWP0002\0

HP Baseboard Management Controller Interface
Device Yes SYSTEM
7.2.3790.6 8/12/2005 Hewlett
Packard Co. oem0.inf Not Available
ACPI\IPI0001\0

Generic Bus Yes SYSTEM
5.2.3790.1830
10/1/2002 (Standard system
devices) machine.inf Not Available
ACPI\HWP0001\0

ACPI Thermal Zone Yes SYSTEM
5.2.3790.1830
10/1/2002 (Standard system
devices) machine.inf Not Available
ACPI\THERMALZONE\THM0

Microsoft ACPI-Compliant System Yes
SYSTEM 5.2.3790.0
10/1/2002 Microsoft acpi.inf
Not Available ACPI_HAL\PNPOC08\0

ACPI IA64-based PC Yes
COMPUTER 5.2.3790.1830
10/1/2002 (Standard computers)
hal.inf Not Available
ROOT\ACPI_HAL\0000
Not Available Not Available Not Available Not
Available Not Available Not Available Not
Available Not Available HTRREE\ROOT\0

[Environment Variables]

Variable Value User Name
ClusterLog C:\WINDOWS\Cluster\cluster.log
<SYSTEM>

```

```

ComSpec %SystemRoot%\system32\cmd.exe
FP_NO_HOST <SYSTEM> NO
<SYSTEM>
HP_SSL_SHARE C:\hp\sslshare\
<SYSTEM>
NUMBER_OF_PROCESSORS 16
<SYSTEM>
OS Windows_NT <SYSTEM>
Path
%SystemRoot%\system32;%Syste
mRoot%;%SystemRoot%\System32\Wbem;%Sy
stemRoot%\syswow64\C:\Program Files
(x86)\Microsoft SQL
Server\80\Tools\Binn\C:\Program Files\Microsoft
SQL Server\90\Tools\bin\;C:\Program Files
(x86)\Microsoft SQL
Server\90\Tools\bin\;C:\Program Files\Microsoft
SQL Server\MSSQL.1\MSSQL\Binn;
<SYSTEM>
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.J
S;.JSE;.WSF;.WSH <SYSTEM>
PROCESSOR_ARCHITECTURE IA64
<SYSTEM>
PROCESSOR_IDENTIFIER ia64 Family 32 Model
0 Stepping 5, GenuineIntel <SYSTEM>
PROCESSOR_LEVEL 32
<SYSTEM>
PROCESSOR_REVISION 0005
<SYSTEM>
TEMP %SystemRoot%\TEMP
<SYSTEM>
TMP %SystemRoot%\TEMP
<SYSTEM>
windir %SystemRoot%
<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 NT
SQLSAPPHIRE\Administrator
TMP %USERPROFILE%\Local
Settings\Temp
SQLSAPPHIRE\Administrator

[Print Jobs]

Document Size Owner Notify
Status Time Submitted
Start Time Until Time Elapsed
Time Pages Printed Job ID
Priority Parameters Driver
Print Processor Host
Print Queue Data Type Name

```

```

[Network Connections]
Local Name Remote Name Type
Status User Name
Z: \\hpwieshare\wie Disk
Persistent Connection
SQLSAPPHIRE\Administrator

[Running Tasks]
Name Path Process ID Priority
Min Working Set Max
Working Set Start Time Version Size
File Date
system idle process Not Available 0
0 Not Available Not
Available Not Available Not
Available Not Available
system Not Available 4 8
0 2826240 Not
Available Not Available Not
Available Not
smss.exe Not Available 1224 11
409600 2826240
7/7/2006 4:22 PM Not
Available Not Available
csrss.exe Not Available 1592 13
Not Available Not Available
7/7/2006 4:23 PM Not
Available Not Available Not
Available
winlogon.exe c:\windows\system32\winlogon.exe
420 13 409600
2826240 7/7/2006 4:23 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 664.50
KB (680,448 bytes) 3/25/2005 4:00 AM

services.exe c:\windows\system32\services.exe
508 9 409600
2826240 7/7/2006 4:23 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 300.00
KB (307,200 bytes) 3/25/2005 4:00 AM

lsass.exe c:\windows\system32\lsass.exe
520 9 409600
2826240 7/7/2006 4:23 PM
5.2.3790.0 (srv03_rtm.030324-
2048) 15.00 KB (15,360 bytes)
3/25/2005 4:00 AM

svchost.exe c:\windows\system32\svchost.exe
768 8 409600
2826240 7/7/2006 4:23 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 36.00 KB
(36,864 bytes) 3/25/2005 4:00 AM

svchost.exe Not Available 932 8
Not Available Not Available
7/7/2006 4:23 PM Not
Available Not Available Not
Available 988 8
Not Available Not Available
7/7/2006 4:23 PM Not
Available Not Available Not
Available 1064 8
Not Available Not Available
7/7/2006 4:23 PM Not
Available Not Available Not
Available

```

```

svchost.exe c:\windows\system32\svchost.exe
1100 8 409600
2826240 7/7/2006 4:23 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 36.00 KB
(36,864 bytes) 3/25/2005 4:00 AM

msdtc.exe Not Available 1500 8
Not Available Not Available
7/7/2006 4:23 PM Not
Available Not Available Not Available
svchost.exe c:\windows\system32\svchost.exe
476 8 409600
2826240 7/7/2006 4:23 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 36.00 KB
(36,864 bytes) 3/25/2005 4:00 AM

wmiprvse.exe Not Available 1036
8 Not Available Not
Available 7/7/2006 4:24 PM Not
Available Not Available Not Available
csrss.exe Not Available 1672 13
Not Available Not Available
7/7/2006 4:25 PM Not
Available Not Available Not Available
winlogon.exe c:\windows\system32\winlogon.exe
1960 13 409600
2826240 7/7/2006 4:25 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 664.50
KB (680,448 bytes) 3/25/2005 4:00 AM

rdpclip.exe c:\windows\system32\rdpclip.exe
888 8 409600
2826240 7/7/2006 4:25 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 211.00
KB (216,064 bytes) 3/25/2005 4:00 AM

explorer.exe c:\windows\explorer.exe 1384
8 409600 2826240
7/7/2006 4:25 PM
6.00.3790.1830
(srv03_sp1_rtm.050324-1447) 1.64 MB
(1,720,320 bytes) 3/25/2005 4:00 AM

logon.scr Not Available 3912 4
Not Available Not Available
7/7/2006 4:35 PM Not
Available Not Available Not Available
iexplore.exe c:\program files (x86)\internet
explorer\iexplore.exe 3244 8
409600 2826240
7/8/2006 10:54 PM
6.00.3790.1830
(srv03_sp1_rtm.050324-1447) 92.00 KB
(94,208 bytes) 3/25/2005 4:00 AM

mmc.exe c:\windows\system32\mmc.exe
4016 8 409600
2826240 7/8/2006 11:39 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 2.51 MB
(2,631,168 bytes) 3/25/2005 4:00 AM

helpctr.exe c:\windows\pchealth\helpctr\binarie
s\helpctr.exe 1004 8 409600
2826240 7/8/2006 11:39 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 2.01 MB
(2,105,344 bytes) 3/25/2005 4:00 AM

```

```

wmiprvse.exe Not Available 4028
8 Not Available Not
Available 7/8/2006 11:39 PM Not
Available Not Available Not Available
c:\windows\pchealth\helpctr\binarie
s\helpsvc.exe 2284 8 409600
2826240 7/8/2006 11:39 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 2.21 MB
(2,320,384 bytes) 3/25/2005 4:00 AM

[Loaded Modules]

Name Version Size File Date
ManufacturerPath
winlogon 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 664.50
KB (680,448 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\winlogon.exe

ntdll 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 1.56 MB
(1,638,400 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\ntdll.dll

kernel32 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 1.87 MB
(1,965,056 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\kernel32.dll

msvcrt 7.0.3790.1830
(srv03_sp1_rtm.050324-1447) 932.00
KB (954,368 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\msvcrt.dll

advapi32 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 1.41 MB
(1,482,752 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\advapi32.dll

rpcrt4 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 2.34 MB
(2,457,600 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\rpcrt4.dll

user32 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 1.41 MB
(1,476,096 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\user32.dll

gdi32 5.2.3790.2542
(srv03_sp1_gdr.051005-1515) 886.00
KB (907,264 bytes) 10/5/2005 1:03 PM
Microsoft Corporation
c:\windows\system32\gdi32.dll

userenv 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 1.49 MB
(1,563,648 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\userenv.dll

```

```

nddeapi 5.2.3790.0 (srv03_rtm.030324-
2048) 39.50 KB (40,448 bytes)
3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\nddeapi.dll

crypt32 5.1.31.3790.1830
(srv03_sp1_rtm.050324-1447) 1.68 MB
(1,759,232 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\crypt32.dll

msasn1 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 179.50
KB (183,808 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\msasn1.dll

secur32 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 186.00
KB (190,464 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\secur32.dll

winsta 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 143.50
KB (146,944 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\winsta.dll

netapi32 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 885.50
KB (906,752 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\netapi32.dll

profmap 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 59.50 KB
(60,928 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\profmap.dll

regapi 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 141.50
KB (144,896 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\regapi.dll

ws2_32 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 248.00
KB (253,952 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\ws2_32.dll

ws2help 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 51.00 KB
(52,224 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\ws2help.dll

msgina 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 1.40 MB
(1,465,344 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\msgina.dll

shsvcs 6.00.3790.1830
(srv03_sp1_rtm.050324-1447) 354.50
KB (363,008 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\shsvcs.dll

```

shlwapi 6.00.3790.2564  
(srv03\_sp1\_gdr.051104-1524) 805.00  
KB (824,320 bytes) 11/30/2005 1:51 PM  
Microsoft Corporation  
c:\windows\system32\shlwapi.dll

sfc 5.2.3790.0 (srv03\_rtm.030324-  
2048) 7.50 KB (7,680 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\sfc.dll

sfc\_os 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 273.00  
KB (279,552 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\sfc\_os.dll

wintrust 5.131.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 459.50  
KB (470,528 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wintrust.dll

imagehlp 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 140.50  
KB (143,872 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\imagehlp.dll

ole32 5.2.3790.2492  
(srv03\_sp1\_gdr.050720-1523) 3.81 MB  
(3,992,064 bytes) 8/29/2005 10:48 AM  
Microsoft Corporation  
c:\windows\system32\ole32.dll

comctl32 6.0 (srv03\_sp1\_rtm.050324-1447)  
2.50 MB (2,622,976 bytes)  
3/24/2005 1:56 AM  
Microsoft Corporation  
c:\windows\winsxs\ia64\_microsoft.  
windows.common-  
controls\_6595b64144ccf1df\_6.0.3790.1830\_x-  
ww\_aa3e736f\comctl32.dll  
version 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 51.00 KB  
(52,224 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\version.dll

winscard 5.2.3790.0 (srv03\_rtm.030324-  
2048) 291.50 KB (298,496 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wincard.dll

wtsapi32 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 54.00 KB  
(55,296 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wtsapi32.dll

sxs 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 1.82 MB  
(1,904,640 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\sxs.dll

shell32 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 12.63  
MB (13,243,392 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\shell32.dll

setupapi 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 1.99 MB  
(2,086,400 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\setupapi.dll

wldap32 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 441.50  
KB (452,096 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wldap32.dll

cscdll 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 207.00  
KB (211,968 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\cscdll.dll

dimntfy 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 51.50 KB  
(52,736 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\dimntfy.dll

wlnotify 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 239.50  
KB (245,248 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wlnotify.dll

winmm 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 428.00  
KB (438,272 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\winmm.dll

winspool 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 400.50  
KB (410,112 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\winspool.drv

mpr 5.2.3790.0 (srv03\_rtm.030324-  
2048) 163.00 KB (166,912 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\mpr.dll

oleaut32 5.2.3790.1830 3.75 MB  
(3,930,624 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\oleaut32.dll

comctl32 5.82 (srv03\_sp1\_rtm.050324-1447)  
1.72 MB (1,806,336 bytes)  
3/24/2005 1:56 AM  
Microsoft Corporation  
c:\windows\winsxs\ia64\_microsoft.  
windows.common-  
controls\_6595b64144ccf1df\_5.82.3790.1830\_x-  
ww\_4aca2dea\comctl32.dll  
uxtheme 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 543.00  
KB (556,032 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\uxtheme.dll

scredir 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 69.00 KB  
(70,656 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\scredir.dll

samlib 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 106.50  
KB (109,056 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\samlib.dll

clbcatq 2001.12.4720.2492  
(srv03\_sp1\_gdr.050720-1523) 1.29 MB  
(1,351,680 bytes) 8/29/2005 10:48 AM  
Microsoft Corporation  
c:\windows\system32\clbcatq.dll

comres 2001.12.4720.0  
(srv03\_rtm.030324-2048) 779.50 KB (798,208  
bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\comres.dll

cscui 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 628.00  
KB (643,072 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\cscui.dll

drprov 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 32.50 KB  
(33,280 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\drprov.dll

rdpsnd 5.2.3790.0 (srv03\_rtm.030324-  
2048) 63.00 KB (64,512 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\rdpsnd.dll

psapi 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 53.00 KB  
(54,272 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\psapi.dll

ntlanman 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 117.00  
KB (119,808 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\ntlanman.dll

netui0 5.2.3790.0 (srv03\_rtm.030324-  
2048) 181.50 KB (185,856 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\netui0.dll

netui1 5.2.3790.0 (srv03\_rtm.030324-  
2048) 482.00 KB (493,568 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\netui1.dll

davclnt 5.2.3790.0 (srv03\_rtm.030324-  
2048) 59.00 KB (60,416 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\davclnt.dll

mprui 5.2.3790.0 (srv03\_rtm.030324-  
2048) 97.00 KB (99,328 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\mprui.dll

netui2 5.2.3790.0 (srv03\_rtm.030324-2048) 761.50 KB (779,776 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\netui2.dll

comdlg32 6.00.3790.1830 (srv03\_sp1\_rtm.050324-1447) 745.50 KB (763,392 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\comdlg32.dll

netmsg 5.2.3790.0 (srv03\_rtm.030324-2048) 177.50 KB (181,760 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\netmsg.dll

msacm32 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 87.00 KB (89,088 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\msacm32.drv

msacm32 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 276.50 KB (283,136 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\msacm32.dll

imaadp32 5.2.3790.0 (srv03\_rtm.030324-2048) 55.00 KB (56,320 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\imaadp32.ac

m msadp32 5.2.3790.0 (srv03\_rtm.030324-2048) 49.00 KB (50,176 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\msadp32.ac

m msg711 5.2.3790.0 (srv03\_rtm.030324-2048) 33.00 KB (33,792 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\msg711.acm

ntmarta 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 366.00 KB (374,784 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\ntmarta.dll

msgsm32 5.2.3790.0 (srv03\_rtm.030324-2048) 66.50 KB (68,096 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\msgsm32.ac

m xpsp2res 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 2.76 MB (2,897,920 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\xpsp2res.dll

tssoft32 1.01 29.00 KB (29,696 bytes) 3/25/2005 4:00 AM DSP GROUP, INC. c:\windows\system32\tssoft32.acm

tsd32 1.03 38.00 KB (38,912 bytes) 3/25/2005 4:00 AM DSP GROUP, INC. c:\windows\system32\tsd32.dll

wbemprox 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 65.00 KB (66,560 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\wbem\wbemprox.dll

plbapi 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 665.50 KB (681,472 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\wbem\wbemplbapi.dll

comn.dll 5.2.3790.0 (srv03\_rtm.030324-2048) 62.50 KB (64,000 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\wbem\wbemcomn.dll

wbemsvc 5.2.3790.0 (srv03\_rtm.030324-2048) 62.50 KB (64,000 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\wbem\wbemsvc.dll

svc.dll 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 1.63 MB (1,710,592 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\wbem\fastprox.dll

ox.dll 6.10.2240.8 941.50 KB (964,096 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\msvcpx60.dll

ntdsapi 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 204.50 KB (209,408 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\ntdsapi.dll

dnsapi 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 428.00 KB (438,272 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\dnsapi.dll

services 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 300.00 KB (307,200 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\services.exe

scserv 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 806.50 KB (825,856 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\scserv.dll

authz 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 220.50 KB (225,792 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\authz.dll

umpnpgmr 5.2.3790.2477 (srv03\_sp1\_gdr.050629-1534) 323.50 KB (331,264 bytes) 6/30/2005 3:06 AM Microsoft Corporation c:\windows\system32\umpnpgmr.dll

ncobjapi 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 121.00 KB (123,904 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\ncobjapi.dll

eventlog 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 180.00 KB (184,320 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\eventlog.dll

lsass 5.2.3790.0 (srv03\_rtm.030324-2048) 15.00 KB (15,360 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\lsass.exe

lsasrv 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 2.07 MB (2,166,784 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\lsasrv.dll

samsrv 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 1.09 MB (1,142,784 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\samsrv.dll

cryptdll 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 68.50 KB (70,144 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\cryptdll.dll

msprvs 5.2.3790.0 (srv03\_rtm.030324-2048) 46.00 KB (47,104 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\msprvs.dll

kerberos 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 932.00 KB (954,368 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\kerberos.dll

msv1\_0 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 380.00 KB (389,120 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\msv1\_0.dll

iphlpapi 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 254.50 KB (260,608 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\iphlpapi.dll

netlogon 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 956.00 KB (978,944 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\netlogon.dll

w32time 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 565.00 KB (578,560 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\w32time.dll

schannel 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 456.00 KB (466,944 bytes) 3/25/2005 4:00 AM Microsoft Corporation c:\windows\system32\schannel.dll

wdigest 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 210.00  
KB (215,040 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wdigest.dll

rsaenh 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 415.98  
KB (425,960 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\rsaenh.dll

rassfm 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 68.50 KB  
(70,144 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\rassfm.dll

kdcsvc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 596.00  
KB (610,304 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\kdcsvc.dll

ntdsa 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 4.04 MB  
(4,239,360 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\ntdsa.dll

esent 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 2.65 MB  
(2,776,064 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\esent.dll

ntdsatq 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 79.50 KB  
(81,408 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\ntdsatq.dll

msocket 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 763.00  
KB (781,312 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\msocket.dll

scecli 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 459.50  
KB (470,528 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\scecli.dll

ws03res 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 792.50  
KB (811,520 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\ws03res.dll

hnetcfg 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 1.04 MB  
(1,094,144 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\hnetcfg.dll

wshtcpip 5.2.3790.0 (srv03\_rtm.030324-  
2048) 38.00 KB (38,912 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wshtcpip.dll

pstorsvc 5.2.3790.0 (srv03\_rtm.030324-  
2048) 56.00 KB (57,344 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\pstorsvc.dll

psbase 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 167.50  
KB (171,520 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\psbase.dll

dssenh 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 363.98  
KB (372,712 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\dssenh.dll

svchost 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 36.00 KB  
(36,864 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\svchost.exe

rpcss 5.2.3790.2492  
(srv03\_sp1\_gdr.050720-1523) 828.00  
KB (847,872 bytes) 8/29/2005 10:48 AM  
Microsoft Corporation  
c:\windows\system32\rpcss.dll

wkssvc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 315.50  
KB (323,072 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wkssvc.dll

wiarp 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 76.00 KB  
(77,824 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wiarp.dll

cryptsvc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 139.00  
KB (142,336 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\cryptsvc.dll

certcli 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 580.00  
KB (593,920 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\certcli.dll

atl 3.00.2282 348.00 KB (356,352  
bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\atl.dll

vssapi 5.2.3790.2476  
(srv03\_sp1\_qfe.050628-1725) 1.57 MB  
(1,642,496 bytes) 6/29/2005 12:49 AM  
Microsoft Corporation  
c:\windows\system32\vssapi.dll

dmserver 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 52.00 KB  
(53,248 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\dmserver.dll

es 2001.12.4720.2492  
(srv03\_sp1\_gdr.050720-1523) 685.50  
KB (701,952 bytes) 8/29/2005 10:48 AM  
Microsoft Corporation  
c:\windows\system32\es.dll

srsvc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 197.50  
KB (202,240 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\srsvc.dll

sacsvr 5.2.3790.0 (srv03\_rtm.030324-  
2048) 27.50 KB (28,160 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\sacsvr.dll

wmisvc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 445.50  
KB (456,192 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wbem\wmisvc.dll

c.dll

trkwks 5.2.3790.0 (srv03\_rtm.030324-  
2048) 246.00 KB (251,904 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\trkwks.dll

sens 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 98.00 KB  
(100,352 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\sens.dll

comsvcs 2001.12.4720.2518  
(srv03\_sp1\_gdr.050826-1537) 3.15 MB  
(3,299,840 bytes) 8/29/2005 10:48 AM  
Microsoft Corporation  
c:\windows\system32\comsvcs.dll

browser 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 200.50  
KB (205,312 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\browser.dll

netrap 5.2.3790.0 (srv03\_rtm.030324-  
2048) 30.00 KB (30,720 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\netrap.dll

wbemcore 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 1.94 MB  
(2,038,784 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wbem\wbemcore.dll

esscli 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 1.01 MB  
(1,057,280 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wbem\esscli.dll

dll

wmiutils 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 302.00  
KB (309,248 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wbem\wmiutils.dll

ls.dll

repdrvfs 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 649.00  
 KB (664,576 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\wbem\repdrv

fs.dll  
 wmioprsvd 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 1.39 MB  
 (1,454,592 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\wbem\wmiopr

vsd.dll  
 wbermess 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 1.02 MB  
 (1,070,080 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\wbem\wberm

ess.dll  
 ncprov 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 146.00  
 KB (149,504 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\wbem\ncprov

.dll  
 netman 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 706.50  
 KB (723,456 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\netman.dll

netshell 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 2.95 MB  
 (3,094,528 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\netshell.dll

rtutils 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 91.50 KB  
 (93,696 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\rtutils.dll

credui 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 311.00  
 KB (318,464 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\credui.dll

clusapi 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 158.00  
 KB (161,792 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\clusapi.dll

mprapi 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 265.00  
 KB (271,360 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\mprapi.dll

activeds 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 581.50  
 KB (595,456 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\activeds.dll

adslidpc 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 326.00  
 KB (333,824 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\adslidpc.dll

rasapi32 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 632.50  
 KB (647,680 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\rasapi32.dll

rasman 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 170.00  
 KB (174,080 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\rasman.dll

tapi32 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 518.00  
 KB (530,432 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\tapi32.dll

wzcsvc 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 818.00  
 KB (837,632 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\wzcsvc.dll

wmi 5.2.3790.0 (srv03\_rtm.030324-  
 2048) 5.00 KB (5,120 bytes)  
 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\wmi.dll

dhcpcsvc 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 299.50  
 KB (306,688 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\dhcpcsvc.dll

wininet 6.00.3790.2564  
 (srv03\_sp1\_gdr.051104-1524) 1.62 MB  
 (1,697,792 bytes) 11/30/2005 1:51 PM  
 Microsoft Corporation  
 c:\windows\system32\wininet.dll

wzcsapi 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 86.50 KB  
 (88,576 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\wzcsapi.dll

rasdlg 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 1.44 MB  
 (1,509,888 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\rasdlg.dll

rasadhlp 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 18.00 KB  
 (18,432 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\rasadhlp.dll

ntlsap 5.2.3790.0 (srv03\_rtm.030324-  
 2048) 14.50 KB (14,848 bytes)  
 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\ntlsapi.dll

netcfgx 5.2.3790.2520  
 (srv03\_sp1\_qfe.050830-1536) 2.01 MB  
 (2,104,320 bytes) 8/30/2005 1:41 PM  
 Microsoft Corporation  
 c:\windows\system32\netcfgx.dll

winipsec 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 83.00 KB  
 (84,992 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\winipsec.dll

xactsrv 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 297.00  
 KB (304,128 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\xactsrv.dll

wbermcons 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 154.50  
 KB (158,208 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\wbem\wberm

cons.dll  
 pchsvc 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 113.00  
 KB (115,712 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\pchealth\helpctr\binarie

s\pchsvc.dll  
 termsrv 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 680.00  
 KB (696,320 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\termsrv.dll

icaapi 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 33.50 KB  
 (34,304 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\icaapi.dll

mstlsapi 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 340.50  
 KB (348,672 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\mstlsapi.dll

rdpwsx 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 325.13  
 KB (332,936 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\rdpwsx.dll

rdpcclip 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 211.00  
 KB (216,064 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\rdpcclip.exe

wsock32 5.2.3790.0 (srv03\_rtm.030324-  
 2048) 23.00 KB (23,552 bytes)  
 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\system32\wsock32.dll

urlmon 6.00.3790.2564  
 (srv03\_sp1\_gdr.051104-1524) 1.53 MB  
 (1,604,608 bytes) 11/30/2005 1:51 PM  
 Microsoft Corporation  
 c:\windows\system32\urlmon.dll

explorer 6.00.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 1.64 MB  
 (1,720,320 bytes) 3/25/2005 4:00 AM  
 Microsoft Corporation  
 c:\windows\explorer.exe

browseui 6.00.3790.2564  
(srv03\_sp1\_gdr.051104-1524) 2.43 MB  
(2,546,688 bytes) 11/30/2005 1:51 PM  
Microsoft Corporation  
c:\windows\system32\browseui.dll

shdocvw 6.00.3790.2580  
(srv03\_sp1\_gdr.051130-1605) 3.51 MB  
(3,679,232 bytes) 11/30/2005 1:51 PM  
Microsoft Corporation  
c:\windows\system32\shdocvw.dll

cryptui 5.131.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 1.11 MB  
(1,159,168 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\cryptui.dll

apphelp 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 301.50  
KB (308,736 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\apphelp.dll

themeui 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 891.00  
KB (912,384 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\themeui.dll

msimg32 5.2.3790.0 (srv03\_rtm.030324-  
2048) 7.00 KB (7,168 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\msimg32.dll

linkinfo 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 46.50 KB  
(47,616 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\linkinfo.dll

ntshrui 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 244.00  
KB (249,856 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\ntshrui.dll

msi 3.1.4000.1830 5.71 MB  
(5,984,768 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\msi.dll

webcheck 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 699.00  
KB (715,776 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\webcheck.dll

stobject 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 175.00  
KB (179,200 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\stobject.dll

batmeter 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 61.00 KB  
(62,464 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\batmeter.dll

powrprof 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 44.00 KB  
(45,056 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\powrprof.dll

browselc 6.00.3790.0 (srv03\_rtm.030324-  
2048) 61.50 KB (62,976 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\browselc.dll

shdoclc 6.00.3790.0 (srv03\_rtm.030324-  
2048) 588.00 KB (602,112 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\shdoclc.dll

actxprxy 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 237.00  
KB (242,688 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\actxprxy.dll

mydocs 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 130.50  
KB (133,632 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\mydocs.dll

iexplore 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 92.00 KB  
(94,208 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\program files (x86)\internet  
explorer\iexplore.exe

wow64 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 487.00  
KB (498,688 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wow64.dll

wow64win 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 567.50  
KB (581,120 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wow64win.dll

wow64cpu 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 44.00 KB  
(45,056 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\wow64cpu.dll

wowia32x 5,3,5021,0 54.00 KB (55,296  
bytes) 3/25/2005 4:00 AM Intel  
Corporation c:\windows\system32\wowia32x.dll

ia32exec 5,3,5338,0 7.13 MB (7,471,104  
bytes) 5/20/2005 2:47 AM Intel  
Corporation c:\windows\system32\ia32exec.bin

mmc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 2.51 MB  
(2,631,168 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\mmc.exe

mfc42u 6.50.4245.0 3.35 MB (3,510,272  
bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\mfc42u.dll

oleacc 4.2.5406.0 (srv03\_rtm.030324-  
2048) 485.00 KB (496,640 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\oleacc.dll

mmcbase 5.2.3790.0 (srv03\_rtm.030324-  
2048) 139.00 KB (142,336 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\mmcbase.dll

mmcmdmgr 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 3.18 MB  
(3,331,584 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\mmcmdmgr.dll

msxml3 8.70.1104.0 3.33 MB (3,488,768  
bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\msxml3.dll

filemgmt 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 928.00  
KB (950,272 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\filemgmt.dll

cfgmgr32 5.2.3790.0 (srv03\_rtm.030324-  
2048) 16.00 KB (16,384 bytes)  
3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\cfgmgr32.dll

snmnsnap 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 565.00  
KB (578,560 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\snmnsnap.dll

servdeps 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 151.50  
KB (155,136 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\servdeps.dll

mmfutil 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 28.00 KB  
(28,672 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\mmfutil.dll

mlang 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 813.50  
KB (833,024 bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\mlang.dll

mshtml 6.00.3790.2577  
(srv03\_sp1\_gdr.051123-1244) 8.86 MB  
(9,290,240 bytes) 11/30/2005 1:51 PM  
Microsoft Corporation  
c:\windows\system32\mshtml.dll

msls31 3.10.349.0 502.50 KB (514,560  
bytes) 3/25/2005 4:00 AM  
Microsoft Corporation  
c:\windows\system32\msls31.dll

```

msimtf 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 586.50
KB (600,576 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\msimtf.dll

msctf 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 983.50
KB (1,007,104 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\msctf.dll

jscript 5.6.0.8827 1.24 MB (1,304,576
bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\jscript.dll

imm32 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 322.50
KB (330,240 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\imm32.dll

mshhtml 6.00.3790.1830
(srv03_sp1_rtm.050324-1447) 1.46 MB
(1,531,392 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\mshhtml.dll

helpctr 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 2.01 MB
(2,105,344 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\pchealth\helpctr\binarie
s\helpctr.exe
hcappres 5.2.3790.0 (srv03_rtm.030324-
2048) 6.00 KB (6,144 bytes)
3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\pchealth\helpctr\binarie
s\hcappres.dll
itss 5.2.3790.2427
(srv03_sp1_gdr.050421-1629) 411.00
KB (420,864 bytes) 4/22/2005 1:17 AM
Microsoft Corporation
c:\windows\system32\itss.dll

pchshell 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 290.50
KB (297,472 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\pchealth\helpctr\binarie
s\pchshell.dll
vbscript 5.6.0.8827 1.07 MB (1,118,208
bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\vbscript.dll

msinfo 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 1.26 MB
(1,321,472 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\pchealth\helpctr\binarie
s\msinfo.dll
riched32 5.2.3790.0 (srv03_rtm.030324-
2048) 5.00 KB (5,120 bytes)
3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\riched32.dll

riched20 5.31.23.12241.38 MB (1,451,008
bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\system32\riched20.dll

```

```

helpsvc 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 2.21 MB
(2,320,384 bytes) 3/25/2005 4:00 AM
Microsoft Corporation
c:\windows\pchealth\helpctr\binarie
s\helpsvc.exe
[Services]

Display Name Name State
Start Mode Service Type Path
Error Control Start Name Tag ID

Application Experience Lookup Service
AeLookupSvc Stopped Disabled
Share Process
c:\windows\system32\svchost.exe -
k netsvcs Normal LocalSystem 0

Alerter Alerter 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
Disabled Share Process
c:\windows\system32\svchost.exe -
k netsvcs Normal LocalSystem 0

ASP.NET State Service aspnet_state Stopped
Manual Own Process
c:\windows\microsoft.net\framework
k64\v2.0.50727\aspnet_state.exe Normal
NT AUTHORITY\NetworkService
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 netsvcs Normal LocalSystem 0

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

.NET Runtime Optimization Service
v2.0.50727_X86
clr_optimization_v2.0.50727_32
Stopped Disabled Own
Process c:\windows\microsoft.net\framework
k\v2.0.50727\mscorlib.exe Ignore
LocalSystem 0

```

```

.NET Runtime Optimization Service
v2.0.50727_I64
clr_optimization_v2.0.50727_64
Stopped Manual Own
Process c:\windows\microsoft.net\framework
k64\v2.0.50727\mscorlib.exe Ignore
COM+ System Application COMSysApp Stopped
Disabled Own Process
c:\windows\system32\dllhost.exe
/processid:{02d4b3f1-fd88-11d1-960d-
00805fc79235} Normal
LocalSystem 0
Cryptographic Services CryptSvc Running
Auto Share Process
c:\windows\system32\svchost.exe -
k netsvcs Normal LocalSystem 0

DCOM Server Process Launcher
DcomLaunch Running Auto
Share Process
c:\windows\system32\svchost.exe -
k dcomlaunch Normal
LocalSystem 0
Distributed File System Dfs Stopped
Manual 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 Stopped 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

Emulex HBAnyware Discovery Emulex
HBAnyware Discovery Stopped Disabled
Own Process "c:\program files
(x86)\hbanyware\hbadiscsrvr.exe" Normal
LocalSystem 0
Emulex HBAnyware SvcMgr Emulex
HBAnyware SvcMgr Stopped Disabled
Own Process "c:\program files
(x86)\hbanyware\hbahsmgr.exe" Normal
LocalSystem 0
Emulex HBAnyware Emulex HBAnyware
Stopped Disabled Own
Process "c:\program files
(x86)\hbanyware\lmsrver.exe" Normal
LocalSystem 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       |          |
|                                  | Auto                              | Share Process |          |
|                                  | c:\windows\system32\svchost.exe - |               |          |
| k netsvcs                        | Normal                            | LocalSystem   | 0        |
| Help and Support                 | helpsvc                           | Running       |          |
|                                  | Manual                            | Share Process |          |
|                                  | c:\windows\system32\svchost.exe - |               |          |
| k netsvcs                        | Normal                            | LocalSystem   | 0        |
| Human Interface Device Access    | HidServ                           | Stopped       | Share    |
|                                  | Stopped                           | Disabled      | Share    |
| Process                          | c:\windows\system32\svchost.exe - |               |          |
| k netsvcs                        | Normal                            | LocalSystem   | 0        |
| HTTP SSL                         | HTTPFilter                        | Stopped       | Manual   |
|                                  | Share Process                     |               |          |
|                                  | c:\windows\system32\ssass.exe     |               |          |
|                                  | Normal                            | LocalSystem   | 0        |
| IAS Jet Database Access          | IASJet                            | Stopped       |          |
|                                  | Manual                            | Share Process |          |
|                                  | c:\windows\system32\svchost.exe - |               |          |
| -k iasjet                        | Normal                            | LocalSystem   | 0        |
| IMAPI CD-Burning COM Service     | ImapiService                      | Stopped       | Disabled |
|                                  | Own Process                       |               |          |
|                                  | c:\windows\system32\imapi.exe     |               |          |
|                                  | Normal                            | LocalSystem   | 0        |
| Intersite Messaging              | IsmServ                           | Stopped       |          |
|                                  | Disabled                          | Own Process   |          |
|                                  | c:\windows\system32\ismserv.exe   |               |          |
|                                  | Normal                            | LocalSystem   | 0        |
| Kerberos Key Distribution Center | kdc                               | Stopped       | Share    |
|                                  | Stopped                           | Disabled      | Share    |
| Process                          | c:\windows\system32\ssass.exe     |               |          |
|                                  | Normal                            | LocalSystem   | 0        |
| Server                           | lanmanserver                      | Running       | 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       | Own      |
|                                  | Stopped                           | Disabled      | Own      |
| Process                          | c:\windows\system32\lssrv.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        |

|  |  |               |          |
|--|--|---------------|----------|
| NetMeeting Remote Desktop Sharing        | mnmrvc   | Stopped       | Disabled |
|  | Own Process  |               |          |
|  | c:\windows\system32\mnmrvc.exe   |               |          |
|  | Normal   | LocalSystem   | 0        |
| Distributed Transaction Coordinator      | MSDTC  | Running       | Auto     |
|  | Running  | Auto          | Own      |
| Process                                  | c:\windows\system32\msdtc.exe  |               |          |
|  | Normal   | NT            |          |
|  | AUTHORITY\NetworkService   |               | 0        |
| SQL Server FullText Search (MSSQLSERVER) | msftesql   | Stopped       | Disabled |
|  | Own Process  |               |          |
|  | "c:\program files\microsoft sql files\microsoft sql server\mssql.1\mssql\bin\msftesql.exe" - |               |          |
| s:mssql.1 -f:mssqlserver                 | Normal   |               |          |
|  | LocalSystem  |               | 0        |
| Windows Installer                        | MSIServer  | Stopped       |          |
|  | Manual   | Share Process |          |
|  | c:\windows\system32\msiexec.exe  |               |          |
| /v                                       | Normal   | LocalSystem   | 0        |
| SQL Server (MSSQLSERVER)                 | MSSQLSERVER  | Stopped       |          |
|  | Manual   | Own Process   |          |
|  | "c:\program files\microsoft sql server\mssql.1\mssql\bin\sqlservr.exe" -                     |               |          |
| smssqlserver                             | Normal   | LocalSystem   | 0        |
| SQL Server Active Directory Helper       | MSSQLServerADHelper  | Stopped       |          |
|  | Disabled   | Own Process   |          |
|  | "c:\program files\microsoft sql server\90\shared\sqladhip90.exe" -                           |               |          |
| server\90\shared\sqladhip90.exe" -       | Normal   |               |          |
|  | NT AUTHORITY\NetworkService  |               | 0        |
| Network DDE                              | NetDDE   | Stopped       | Disabled |
|  | Share Process  |               |          |
|  | c:\windows\system32\netdde.exe   |               |          |
|  | Normal   | LocalSystem   | 0        |
| Network DDE DSDM                         | NetDDEdsdm   | Stopped       |          |
|  | Disabled   | Share Process |          |
|  | c:\windows\system32\netdde.exe   |               |          |
|  | Normal   | LocalSystem   | 0        |
| Net Logon                                | Netlogon   | Stopped       | Manual   |
|  | Share Process  |               |          |
|  | c:\windows\system32\lssass.exe   |               |          |
|  | Normal   | LocalSystem   | 0        |
| Network Connections                      | Netman   | Running       |          |
|  | Manual   | Share Process |          |
|  | c:\windows\system32\svchost.exe -  |               |          |
| k netsvcs                                | Normal   | LocalSystem   | 0        |
| Network Location Awareness (NLA)         | Nla  | Running       | Manual   |
|  | Running  | Manual        | Share    |
| Process                                  | c:\windows\system32\svchost.exe -  |               |          |
| k netsvcs                                | Normal   | LocalSystem   | 0        |
| File Replication                         | Ntfrs  | Stopped       |          |
|  | Manual   | Own Process   |          |
|  | c:\windows\system32\ntfrs.exe  |               |          |
|  | Ignore   | LocalSystem   | 0        |
| NT LM Security Support Provider          | NtLmSsp  | Stopped       | Manual   |
|  | Stopped  | Manual        | Share    |
| Process                                  | c:\windows\system32\lssass.exe   |               |          |
|  | Normal   | LocalSystem   | 0        |

|                                       |  |               |        |
|---------------------------------------|--|---------------|--------|
| Removable Storage                     | NtmsSvc  | Stopped       |        |
|                                       | Manual   | Share Process |        |
|                                       | c:\windows\system32\svchost.exe -  |               |        |
| k netsvcs                             | Normal   | LocalSystem   | 0      |
| Office Source Engine                  | ose  | Stopped       |        |
|                                       | Manual   | Own Process   |        |
|                                       | "c:\program files (x86)\common files\microsoft shared\source engine\ose.exe" - |               |        |
|                                       | Normal   | LocalSystem   | 0      |
| Plug and Play                         | PlugPlay   | Running       | Auto   |
|                                       | Share Process  |               |        |
|                                       | c:\windows\system32\services.exe   |               |        |
|                                       | Normal   | LocalSystem   | 0      |
| IPSEC Services                        | PolicyAgent  | Stopped       |        |
|                                       | Disabled   | Share Process |        |
|                                       | c:\windows\system32\ssass.exe  |               |        |
|                                       | Normal   | LocalSystem   | 0      |
| Protected Storage                     | ProtectedStorage   | Running       | Auto   |
|                                       | Running  | Auto          | Share  |
| Process                               | c:\windows\system32\lssass.exe   |               |        |
|                                       | Normal   | LocalSystem   | 0      |
| 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 |
|                                       | Stopped  | Manual        | Share  |
| Process                               | c:\windows\system32\svchost.exe -  |               |        |
| k 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       | Share  |
|                                       | Stopped  | Disabled      | Share  |
| Process                               | c:\windows\system32\svchost.exe -  |               |        |
| k regsvc                              | Normal   | NT            |        |
|                                       | AUTHORITY\LocalService   |               | 0      |
| Remote Procedure Call (RPC) Locator   | RpclLocator  | Stopped       | Manual |
|                                       | Own Process  |               |        |
|                                       | c:\windows\system32\locator.exe  |               |        |
|                                       | Normal   | NT            |        |
|                                       | AUTHORITY\NetworkService   |               | 0      |
| Remote Procedure Call (RPC)           | RpcSs  | Running       | Auto   |
|                                       | Running  | Auto          | Share  |
| Process                               | c:\windows\system32\svchost.exe -  |               |        |
| k rpcss                               | Normal   | NT            |        |
|                                       | AUTHORITY\NetworkService   |               | 0      |
| Resultant Set of Policy Provider      | RSOProv  | Stopped       | Manual |
|                                       | Share Process  |               |        |
|                                       | c:\windows\system32\rsopprov.exe   |               |        |
|                                       | Normal   | LocalSystem   | 0      |

Special Administration Console Helper saccsvr Running Manual Share  
 Process c:\windows\system32\svchost.exe -  
 k netsvcs Normal LocalSystem 0

Security Accounts ManagerSamSs Running  
 Auto Share Process  
 c:\windows\system32\sass.exe  
 Normal LocalSystem 0

Smart Card SCardSvr Stopped Manual  
 Share Process  
 c:\windows\system32\scardsvr.exe  
 Ignore NT  
 AUTHORITY\LocalService 0

Task Scheduler Disabled Schedule Stopped  
 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

Windows Firewall/Internet Connection Sharing  
 (ICS) SharedAccess Stopped  
 Disabled 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 netsvcs Ignore LocalSystem 0

SNMP Service SNMP Stopped  
 Disabled Own Process  
 c:\windows\system32\snmp.exe  
 Normal LocalSystem 0

SNMP Trap Service SNMPTRAP Stopped  
 Disabled Own Process  
 c:\windows\system32\snmptrap.exe  
 Normal NT  
 AUTHORITY\LocalService 0

Print Spooler Spooler Stopped Disabled  
 Own Process  
 c:\windows\system32\spoolsv.exe  
 Normal LocalSystem 0

SQL Server Browser SQLBrowser Stopped  
 Disabled Own Process  
 "c:\program files\microsoft sql  
 server\90\shared\sqlbrowser.exe"  
 LocalSystem 0 Normal

SQL Server Agent (MSSQLSERVER)  
 SQLSERVERAGENT Stopped  
 Manual Own Process  
 "c:\program files\microsoft sql  
 server\mssql.1\mssql\bin\sqlagent90.exe" -i  
 mssqlserver Normal LocalSystem 0

SQL Server VSS Writer SQLWriter Stopped  
 Manual Own Process  
 "c:\program files\microsoft sql  
 server\90\shared\sqlwriter.exe"  
 LocalSystem 0 Normal

Windows Image Acquisition (WIA) stisvc  
 Stopped Disabled Share  
 Process c:\windows\system32\svchost.exe -  
 k imgsvc Normal NT  
 AUTHORITY\LocalService 0

Microsoft Software Shadow Copy Provider  
 swprv Stopped Manual  
 Own Process  
 c:\windows\system32\svchost.exe -  
 k swprv Normal LocalSystem 0

Performance Logs and Alerts  
 SysmonLog Stopped Auto  
 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 termsvcs Normal LocalSystem 0

Themes Themes Stopped Disabled  
 Share Process  
 c:\windows\system32\svchost.exe -  
 k netsvcs Normal LocalSystem 0

Distributed Link Tracking Server TrkSvr  
 Stopped Disabled Share  
 Process c:\windows\system32\svchost.exe -  
 k netsvcs Normal LocalSystem 0

Distributed Link Tracking Client TrkWks  
 Running Auto Share  
 Process c:\windows\system32\svchost.exe -  
 k netsvcs Normal LocalSystem 0

Terminal Services Session Directory Tssdis  
 Stopped Disabled Own  
 Process c:\windows\system32\tssdis.exe  
 Normal LocalSystem 0

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\vds.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 localservice Normal NT  
 AUTHORITY\LocalService 0

WebClient WebClient Stopped Disabled  
 Share Process  
 c:\windows\system32\svchost.exe -  
 k localservice Normal NT  
 AUTHORITY\LocalService 0

WinHTTP Web Proxy Auto-Discovery Service  
 WinHttpAutoProxySvc Stopped  
 Manual Share Process  
 c:\windows\system32\svchost.exe -  
 k localservice Normal NT

Windows Management Instrumentation  
 Wmi Wmi Stopped Manual  
 Share Process  
 c:\windows\system32\svchost.exe -  
 k 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\wmiap  
 srv.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

Network Provisioning Service xmlprov  
 Stopped Manual Share  
 Process c:\windows\system32\svchost.exe -  
 k netsvcs Normal LocalSystem 0

[Program Groups]

| Group Name                 | Name                                   | User Name |  |
|----------------------------|--|-----------|--|
| Accessories                | Default User:Accessories               | Default   |  |
| Accessories\Accessibility  | Default User:Accessories\Accessibility | Default   |  |
| Accessories\Entertainment  | Default User:Accessories\Entertainment | Default   |  |
| Accessories\Startup        | Default User:Startup                   | Default   |  |
| Accessories                | All Users:Accessories                  | All Users |  |
| Accessories\Accessibility  | All Users:Accessories\Accessibility    | All Users |  |
| Accessories\Communications | All Users:Accessories\Communications   | All Users |  |
| Accessories\Entertainment  | All Users:Accessories\Entertainment    | All Users |  |
| Accessories\System Tools   | All Users:Accessories\System Tools     | All Users |  |
| Administrative Tools       | All Users:Administrative Tools         | All Users |  |
| Emulex                     | All Users:Emulex                       | All Users |  |
| HP System Tools            | All Users:HP System Tools              | All Users |  |



|               |   |          |
|---------------|---|----------|
| imgutil.dll   | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 114 KB   |
| imgutil.dll   | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation     | 38 KB    |
| inetcpl.cpl   | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 666 KB   |
| inetcpl.cpl   | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation     | 358 KB   |
| inetcplc.dll  | 6.0.3790.0 108 KB<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation |          |
| inetcplc.dll  | 6.0.3790.0 109 KB<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation |          |
| inseng.dll    | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 276 KB   |
| inseng.dll    | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation     | 94 KB    |
| mlang.dll     | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 814 KB   |
| mlang.dll     | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation     | 578 KB   |
| msencode.dll  | 2002.10.4.0 112 KB<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64 Not                     |          |
| Available     |   |          |
| mshta.exe     | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 68 KB    |
| mshta.exe     | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation     | 30 KB    |
| mshtml.dll    | 6.0.3790.2577<br>11/30/2005 2:51:44 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation    | 9,073 KB |
| mshtml.dll    | 6.0.3790.2577<br>11/30/2005 2:51:44 PM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation    | 3,040 KB |
| mshtml.tlb    | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 1,319 KB |
| mshtml.tlb    | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation     | 1,320 KB |
| mshtmlmed.dll | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 1,496 KB |

|               |  |          |
|---------------|--|----------|
| mshtmlmed.dll | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation    | 455 KB   |
| mshtmlr.dll   | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation    | 56 KB    |
| mshtmlr.dll   | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation    | 56 KB    |
| msident.dll   | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation    | 144 KB   |
| msident.dll   | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation    | 48 KB    |
| msidntld.dll  | 6.0.3790.0 14 KB<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation |          |
| msidntld.dll  | 6.0.3790.0 15 KB<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation |          |
| msieftp.dll   | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation    | 579 KB   |
| msieftp.dll   | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation    | 244 KB   |
| msrating.dll  | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation    | 415 KB   |
| msrating.dll  | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation    | 144 KB   |
| mstime.dll    | 6.0.3790.2564<br>11/30/2005 2:51:44 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation   | 1,800 KB |
| mstime.dll    | 6.0.3790.2564<br>11/30/2005 2:51:44 PM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation   | 524 KB   |
| occache.dll   | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation    | 218 KB   |
| occache.dll   | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation    | 94 KB    |
| proctexe.ocx  | 6.3.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64 Intel<br>Corporation        | 83 KB    |
| sendmail.dll  | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation    | 109 KB   |
| sendmail.dll  | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation    | 56 KB    |

|              |   |          |
|--------------|---|----------|
| shdoclc.dll  | 6.0.3790.0 588 KB<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation |          |
| shdoclc.dll  | 6.0.3790.0 588 KB<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation |          |
| shdocvw.dll  | 6.0.3790.2580<br>11/30/2005 2:51:44 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation    | 3,593 KB |
| shdocvw.dll  | 6.0.3790.2580<br>11/30/2005 2:51:44 PM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation    | 1,478 KB |
| shfolder.dll | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 43 KB    |
| shfolder.dll | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation     | 25 KB    |
| shlwapi.dll  | 6.0.3790.2564<br>11/30/2005 2:51:44 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation    | 805 KB   |
| shlwapi.dll  | 6.0.3790.2564<br>11/30/2005 2:51:44 PM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation    | 314 KB   |
| tdc.ocx      | 1.3.0.3130 177 KB<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation |          |
| tdc.ocx      | 1.3.0.3130 58 KB<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation  |          |
| url.dll      | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 49 KB    |
| url.dll      | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation     | 37 KB    |
| urlmon.dll   | 6.0.3790.2564<br>11/30/2005 2:51:44 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation    | 1,567 KB |
| urlmon.dll   | 6.0.3790.2564<br>11/30/2005 2:51:44 PM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation    | 675 KB   |
| webcheck.dll | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 699 KB   |
| webcheck.dll | 6.0.3790.1830<br>3/25/2005 5:00:00 AM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation     | 273 KB   |
| wininet.dll  | 6.0.3790.2564<br>11/30/2005 2:51:44 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation    | 1,658 KB |
| wininet.dll  | 6.0.3790.2564<br>11/30/2005 2:51:44 PM<br>C:\WINDOWS\SysWOW64<br>Microsoft Corporation    | 647 KB   |

[Connectivity]

|                       |            |
|-----------------------|------------|
| Item                  | Value      |
| Connection Preference | Never dial |

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                | Not Available   |
| Available Disk Space            | Not Available   |
| Maximum Cache Size              | Not Available   |
| Available Cache Size            | Not Available   |

[List of Objects]

|  |          |
|--|----------|
| Program File Status                    | CodeBase |
| No cached object information available |          |

[Content]

[ Following are sub-categories of this main category ]  
[Summary]

|                 |          |
|-----------------|----------|
| Item            | Value    |
| Content Advisor | Disabled |

[Personal Certificates]

|   |           |           |
|---|-----------|-----------|
| Issued To                                     | Issued By | Validity  |
|   | Signature | Algorithm |
| No personal certificate information available |           |           |

[Other People Certificates]

|   |           |           |
|---|-----------|-----------|
| Issued To   | Issued By | Validity  |
|   | Signature | Algorithm |
| No other people certificate information available |           |           |

[Publishers]

|                                    |  |
|------------------------------------|--|
| Name                               |  |
| No publisher information available |  |

[Security]

|                  |                |
|------------------|----------------|
| Zone             | Security Level |
| My Computer      | Custom         |
| Local intranet   | Custom         |
| Trusted sites    | Low            |
| Internet         | Medium         |
| Restricted sites | Custom         |

**Microsoft SQL Server 2005 Startup Parameters**

```
sqlservr.exe -c -x -T661 -T677 -T697 -T827 -T834 -T1211 -T3502 -T8011 -T8012 -T8018 -T8020 -T8193 -T8710 -T8744 -T9259
```

```
where
rem ** -c - Run as console app
rem ** -x - Disable stats
rem ** -T661 - Disable the ghost record removal process
rem ** -T677 - Disable bunch of checks in AM. Available in chk/fre, not golden
rem ** -T697 - Make schema latch superlatch
rem ** -T827 - Display latch promotions
rem ** -T834 - Large pages
rem ** -T1211 - Disable lock escalation
rem ** -T3502 - Send checkpoint state changes to errorlog
rem ** -T8011 - Disable ring buffer for resource monitor
rem ** -T8012 - Disable ring buffer for schedulers
rem ** -T8018 - Disable exception ring buffer
rem ** -T8020 - Disable working set trimming
rem ** -T8193 - Enable large pages support
rem ** -T8710 - disable HP spools (2000 vs 1w bug)
rem ** -T8744 - Disable pre-fetch
rem ** -T9259 - prevent pullup of projects
```

**Microsoft SQL Server TCP and Soft Numa Config**

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration]
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node0]
"CPUMask"=dword:00000003
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node1]
"CPUMask"=dword:0000000C
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node2]
"CPUMask"=dword:00000030
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node3]
```

```
"CPUMask"=dword:000000C0
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node4]
"CPUMask"=dword:00000300
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node5]
"CPUMask"=dword:00000c00
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node6]
"CPUMask"=dword:00003000
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node7]
"CPUMask"=dword:0000c000
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IPAll]
"TcpPort"="1436[0x1],1437[0x2],1438[0x4],1439[0x8],1440[0x10],1441[0x20],1442[0x40],1443[0x80]"
"TcpDynamicPorts"=""
"DisplayName"="Any IP Address"
```

**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.

**LP1050 Driver Settings**

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\elxstor]
```

```
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000000
"Tag"=dword:00000027
"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,00,65,00,6c,00,78,00,73,00,74,00,6f,00,72,00,2e,00,73,00,79,00,73,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\elxstor\Parameters]
"BusType"=dword:00000006
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\elxstor\Parameters\Device]
```

```
"DriverParameter"="NodeTimeOut=10;LinkTimeOut=40;QueueTarget=1;EmulexOption=0x7cbd30ca;"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\exstor\Parameters\Device0]
"DriverParameter"="CoalesceRspCnt=0;NodeTimeOut=10;LinkTimeOut=40;QueueTarget=1;"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\exstor\Parameters\Device1]
"DriverParameter"="CoalesceRspCnt=0;NodeTimeOut=10;LinkTimeOut=40;QueueTarget=1;"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\exstor\Parameters\PnpInterface]
"5"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\exstor\Enum]
"0"="PCI\VEN_10DF&DEV_F0A5&SUBSYS_F0A510DF&REV_01\4&2c178b65&0&08"
"Count"=dword:00000002
"NextInstance"=dword:00000002
"1"="PCI\VEN_10DF&DEV_F0A5&SUBSYS_F0A510DF&REV_01\4&2c178b65&0&09"
```

### QLDIRECT Driver Settings

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qldirect]
"ERRORCONTROL"=dword:00000001
"TYPE"=dword:00000001
"GROUP"="qldirect"
"START"=dword:00000002
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qldirect\Parameters]
"SRBLISTSIZE"=dword:00000800
"FLAGS"=dword:00000000
"MAXPATHSPERDEVICE"=dword:00000001
"INSPECTIONINTERVAL"=dword:00000258
"OPTIMIZATION"=dword:00000000
"MAXRETRIESPERPATH"=dword:00000003
"MAXRETRIESPERIO"=dword:00000008
"PerCpuData"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qldirect\Enum]
"0"="Root\LEGACY_QLDIRECT\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

### Intel PRO/1000 MT Dual Port Server Adapter

All default, except Interrupt Moderation = Extreme

## C.1 Microsoft SQL Server 8.0 Configuration Parameters

| name                           | minimum     | maximum    | config_value | run_value |
|--------------------------------|-------------|------------|--------------|-----------|
| Ad Hoc Distributed Queries     | 0           | 1          | 0            | 0         |
| affinity I/O mask              | -2147483648 | 2147483647 | 0            | 0         |
| affinity mask                  | -2147483648 | 2147483647 | 65535        | 65535     |
| affinity64 I/O mask            | -2147483648 | 2147483647 | 0            | 0         |
| affinity64 mask                | -2147483648 | 2147483647 | 0            | 0         |
| Agent XPs                      | 0           | 1          | 0            | 1         |
| allow updates                  | 0           | 1          | 0            | 1         |
| awe enabled                    | 0           | 1          | 0            | 1         |
| blocked process threshold      | 0           | 86400      | 0            | 0         |
| c2 audit mode                  | 0           | 1          | 0            | 1         |
| clr enabled                    | 0           | 1          | 0            | 1         |
| cost threshold for parallelism | 0           | 32767      | 5            | 5         |
| cross db ownership chaining    | 0           | 1          | 0            | 0         |
| cursor threshold               | -1          | -1         | -1           | -1        |
| Database Mail XPs              | 0           | 1          | 0            | 1         |
| default full-text language     | 0           | 2147483647 | 1033         | 1033      |
| default language               | 0           | 9999       | 0            | 0         |
| default trace enabled          | 0           | 1          | 0            | 1         |
| disallow results from triggers | 0           | 1          | 0            | 0         |
| fill factor (%)                | 0           | 100        | 0            | 100       |
| ft crawl bandwidth (max)       | 0           | 32767      | 100          | 100       |
| ft crawl bandwidth (min)       | 0           | 32767      | 0            | 0         |
| ft notify bandwidth (max)      | 0           | 32767      | 100          | 100       |
| ft notify bandwidth (min)      | 0           | 32767      | 0            | 0         |
| in-doubt xact resolution       | 0           | 2          | 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           | 64         | 1            | 1         |
| max full-text crawl range      | 0           | 256        | 4            | 4         |

|                           |      |            |        |        |
|---------------------------|------|------------|--------|--------|
| max server memory (MB)    | 0    | 16         | 180000 | 180000 |
| max text rep size (KB)    | 0    | 2147483647 | 65536  | 65536  |
| max worker threads        | 128  | 32767      | 800    | 800    |
| media retention           | 0    | 365        | 0      | 0      |
| min memory per query (KB) | 512  | 2147483647 | 1024   | 1024   |
| min server memory (MB)    | 0    | 2147483647 | 0      | 16     |
| nested triggers           | 0    | 1          | 1      | 1      |
| network packet size (B)   | 512  | 32767      | 4096   | 4096   |
| Ole Automation Procedures | 0    | 1          | 0      | 0      |
| open objects              | 0    | 2147483647 | 0      | 0      |
| PH timeout (s)            | 1    | 3600       | 60     | 60     |
| precompute rank           | 0    | 1          | 0      | 0      |
| priority boost            | 0    | 1          | 1      | 1      |
| query governor cost limit | 0    | 2147483647 | 0      | 0      |
| query wait (s)            | -1   | 2147483647 | -1     | -1     |
| recovery interval (min)   | 0    | 32767      | 32767  | 32767  |
| remote access             | 0    | 1          | 1      | 1      |
| remote admin connections  | 0    | 1          | 0      | 0      |
| remote login timeout (s)  | 0    | 2147483647 | 20     | 20     |
| remote proc trans         | 0    | 1          | 0      | 0      |
| remote query timeout (s)  | 0    | 2147483647 | 600    | 600    |
| Replication XPs           | 0    | 1          | 0      | 0      |
| scan for startup procs    | 0    | 1          | 0      | 0      |
| server trigger recursion  | 0    | 1          | 1      | 1      |
| set working set size      | 0    | 1          | 0      | 0      |
| show advanced options     | 0    | 1          | 1      | 1      |
| SMO and DMO XPs           | 0    | 1          | 1      | 1      |
| SQL Mail XPs              | 0    | 1          | 0      | 0      |
| transform noise words     | 0    | 1          | 0      | 0      |
| two digit year cutoff     | 1753 | 9999       | 2049   | 2049   |
| user connections          | 0    | 32767      | 0      | 0      |
| user options              | 0    | 32767      | 0      | 32767  |
| Web Assistant Procedures  | 0    | 1          | 0      | 0      |
| xp_cmdshell               | 0    | 1          | 0      | 0      |

## C.2 Client System Configuration Parameters

### Client Windows Server 2003 Settings

System Information report written at: 07/08/06 23:43:42

System Name: DL18  
[System Summary]

| Item    | Value   |
|---------|---|
| OS Name | Microsoft(R) Windows(R) Server 2003, Standard Edition |
| Version | 5.2.3790 Service Pack 1 Build 3790                    |

|                      |                       |
|----------------------|-----------------------|
| Other OS Description | Not Available         |
| OS Manufacturer      | Microsoft Corporation |

|                     |                   |
|---------------------|-------------------|
| System Name         | DL18              |
| System Manufacturer | HP                |
| System Model        | ProLiant DL140 G2 |

System Type X86-based PC

|                   |   |
|-------------------|---|
| Processor         | x86 Family 15 Model 4 Stepping 3 GenuineIntel ~3600 Mhz |
| Processor         | x86 Family 15 Model 4 Stepping 3 GenuineIntel ~3600 Mhz |
| Processor         | x86 Family 15 Model 4 Stepping 3 GenuineIntel ~3600 Mhz |
| Processor         | x86 Family 15 Model 4 Stepping 3 GenuineIntel ~3600 Mhz |
| BIOS Version/Date | HP 1.14, 9/13/2005                                      |

|                   |            |
|-------------------|------------|
| SMBIOS Version    | 2.33       |
| Windows Directory | C:\WINDOWS |

System Directory C:\WINDOWS\system32

|                            |   |
|----------------------------|---|
| Boot Device                | \Device\HarddiskVolume1                               |
| Locale                     | United States   |
| Hardware Abstraction Layer | Version = "5.2.3790.1830 (srv03_sp1_rtm.050324-1447)" |

|                           |                       |
|---------------------------|-----------------------|
| User Name                 | Not Available         |
| Time Zone                 | Pacific Daylight Time |
| Total Physical Memory     | 1,022.93 MB           |
| Available Physical Memory | 657.33 MB             |
| Total Virtual Memory      | 2.91 GB               |
| Available Virtual Memory  | 2.63 GB               |
| Page File Space           | 2.00 GB               |
| Page File                 | C:\pagefile.sys       |

[Hardware Resources]

[Conflicts/Sharing]

| Resource                       | Device                          | Status |
|--------------------------------|---------------------------------|--------|
| I/O Port 0x00000000-0x00000CF7 | PCI bus                         | OK     |
| I/O Port 0x00000000-0x00000CF7 | Direct memory access controller | OK     |

|                                       |   |    |
|---------------------------------------|---|----|
| Memory Address 0xDA000000-0xDBFFFFFFF | Intel(R) 6700PXH PCI Express-to-PCI Bridge A - 0329 | OK |
|---------------------------------------|---|----|

|                                       |                                     |    |
|---------------------------------------|-------------------------------------|----|
| Memory Address 0xDA000000-0xDBFFFFFFF | Broadcom BCM5706C NetXtreme II GigE | OK |
|---------------------------------------|-------------------------------------|----|

|  |  |    |
|--|--|----|
| Memory Address 0xD8100000-0xD81FFFFFFF | Intel(R) E7525/E7520 PCI Express Root Port B0 - 3597 | OK |
| Memory Address 0xD8100000-0xD81FFFFFFF | Broadcom NetXtreme Gigabit Ethernet #3               | OK |

|  |  |    |
|--|--|----|
| Memory Address 0xD8200000-0xD82FFFFFFF | Intel(R) E7520 PCI Express Root Port B1 - 3598 | OK |
| Memory Address 0xD8200000-0xD82FFFFFFF | Broadcom NetXtreme Gigabit Ethernet #4         | OK |

|        |  |    |
|--------|--|----|
| IRQ 16 | Intel(R) E7525/E7520/E7320 PCI Express Root Port A0 - 3595 | OK |
| IRQ 16 | Intel(R) E7525/E7520 PCI Express Root Port B0 - 3597       | OK |
| IRQ 16 | Broadcom NetXtreme Gigabit Ethernet #3                     | OK |
| IRQ 16 | Intel(R) E7520 PCI Express Root Port B1 - 3598             | OK |
| IRQ 16 | Broadcom NetXtreme Gigabit Ethernet #4                     | OK |
| IRQ 16 | Intel(R) E7520 PCI Express Root Port C0 - 3599             | OK |
| IRQ 16 | Broadcom BCM5706C NetXtreme II GigE                        | OK |
| IRQ 16 | Intel(R) 82801EB USB Universal Host Controller - 24D2      | OK |
| IRQ 16 | RAGE XL PCI Family (Microsoft Corporation)                 | OK |

|                                 |         |    |
|---------------------------------|---------|----|
| Memory Address 0xA0000-0xBFFFFF | PCI bus | OK |
|---------------------------------|---------|----|

|                                 |  |    |
|---------------------------------|--|----|
| Memory Address 0xA0000-0xBFFFFF | RAGE XL PCI Family (Microsoft Corporation) | OK |
|---------------------------------|--|----|

|                                       |   |    |
|---------------------------------------|---|----|
| Memory Address 0xD8300000-0xDBFFFFFFF | Intel(R) E7520 PCI Express Root Port C0 - 3599          | OK |
| Memory Address 0xD8300000-0xDBFFFFFFF | Intel(R) 6700PXH I/OxAPIC Interrupt Controller A - 0326 | OK |

[DMA]

| Resource  | Device                          | Status |
|-----------|---------------------------------|--------|
| Channel 4 | Direct memory access controller | OK     |

[Forced Hardware]

| Device | PNP Device ID |
|--------|---------------|
|        |               |

[I/O]

| Resource              | Device  | Status |
|-----------------------|---|--------|
| 0x00000000-0x00000CF7 | PCI bus   | OK     |
| 0x00000000-0x00000CF7 | Direct memory access controller                       | OK     |
| 0x00000D00-0x0000FFFF | PCI bus   | OK     |
| 0x00001400-0x0000141F | Intel(R) 82801EB USB Universal Host Controller - 24D2 | OK     |
| 0x00001420-0x0000143F | Intel(R) 82801EB USB Universal Host Controller - 24D4 | OK     |

|                       |  |    |
|-----------------------|--|----|
| 0x00002000-0x000020FF | RAGE XL PCI Family (Microsoft Corporation) | OK |
| 0x000003C0-0x000003DF | RAGE XL PCI Family (Microsoft Corporation) | OK |
| 0x00000A79-0x00000A79 | ISAPNP Read Data Port                      | OK |
| 0x00000279-0x00000279 | ISAPNP Read Data Port                      | OK |

|                       |                       |    |
|-----------------------|-----------------------|----|
| 0x00000274-0x00000277 | ISAPNP Read Data Port | OK |
|-----------------------|-----------------------|----|

|                       |                                 |    |
|-----------------------|---------------------------------|----|
| 0x00000010-0x0000001F | Motherboard resources           | OK |
| 0x00000024-0x00000025 | Motherboard resources           | OK |
| 0x00000028-0x00000029 | Motherboard resources           | OK |
| 0x0000002C-0x0000002D | Motherboard resources           | OK |
| 0x0000002E-0x0000002F | Motherboard resources           | OK |
| 0x00000030-0x00000031 | Motherboard resources           | OK |
| 0x00000034-0x00000035 | Motherboard resources           | OK |
| 0x00000038-0x00000039 | Motherboard resources           | OK |
| 0x0000003C-0x0000003D | Motherboard resources           | OK |
| 0x00000050-0x00000053 | Motherboard resources           | OK |
| 0x00000072-0x00000077 | Motherboard resources           | OK |
| 0x00000080-0x00000080 | Motherboard resources           | OK |
| 0x00000090-0x0000009F | Motherboard resources           | OK |
| 0x000000A4-0x000000A5 | Motherboard resources           | OK |
| 0x000000A8-0x000000A9 | Motherboard resources           | OK |
| 0x000000AC-0x000000AD | Motherboard resources           | OK |
| 0x000000B0-0x000000B5 | Motherboard resources           | OK |
| 0x000000B8-0x000000B9 | Motherboard resources           | OK |
| 0x000000BC-0x000000BD | Motherboard resources           | OK |
| 0x000004D0-0x000004D1 | Motherboard resources           | OK |
| 0x00001000-0x0000107F | Motherboard resources           | OK |
| 0x00001180-0x000011BF | Motherboard resources           | OK |
| 0x0000FE00-0x0000FE00 | Motherboard resources           | OK |
| 0x0000FE10-0x0000FE11 | Motherboard resources           | OK |
| 0x00000600-0x0000067F | Motherboard resources           | OK |
| 0x00000CA2-0x00000CA5 | Motherboard resources           | OK |
| 0x00000062-0x00000062 | Motherboard resources           | OK |
| 0x00000066-0x00000066 | Motherboard resources           | OK |
| 0x00000081-0x0000008F | Direct memory access controller | OK |
| 0x000000C0-0x000000DF | Direct memory access controller | OK |
| 0x000000F0-0x000000FE | Numeric data processor          | OK |

```

0x00000020-0x00000021 Programmable
interrupt controller OK
0x000000A0-0x000000A1 Programmable
interrupt controller OK
0x00000070-0x00000071 System CMOS/real
time clock OK
0x00000061-0x00000061 System speaker
OK
0x00000040-0x00000043 System timer OK

0x00000060-0x00000060 Standard 101/102-Key
or Microsoft Natural PS/2 Keyboard OK

0x00000064-0x00000064 Standard 101/102-Key
or Microsoft Natural PS/2 Keyboard OK

0x000003F8-0x000003FF Communications Port
(COM1) OK
0x00001470-0x0000147F Intel(R) 82801EB
Ultra ATA Storage Controllers OK

0x000001F0-0x000001F7 Primary IDE Channel
OK
0x000003F6-0x000003F6 Primary IDE Channel
OK
0x00000170-0x00000177 Secondary IDE
Channel OK
0x00000376-0x00000376 Secondary IDE
Channel OK
0x00001440-0x0000145F Intel(R) 82801EB
SMBus Controller - 24D3 OK

[IRQs]

Resource Device Status
IRQ 9 Microsoft ACPI-Compliant System
OK
IRQ 16 Intel(R) E7525/E7520/E7320 PCI
Express Root Port A0 - 3595 OK

IRQ 16 Intel(R) E7525/E7520 PCI Express
Root Port B0 - 3597 OK
IRQ 16 Broadcom NetXtreme Gigabit
Ethernet #3 OK
IRQ 16 Intel(R) E7520 PCI Express Root
Port B1 - 3598 OK
IRQ 16 Broadcom NetXtreme Gigabit
Ethernet #4 OK
IRQ 16 Intel(R) E7520 PCI Express Root
Port C0 - 3599 OK
IRQ 16 Broadcom BCM5706C NetXtreme II
GigE OK
IRQ 16 Intel(R) 82801EB USB Universal
Host Controller - 24D2 OK
IRQ 16 RAGE XL PCI Family (Microsoft
Corporation) OK
IRQ 19 Intel(R) 82801EB USB Universal
Host Controller - 24D4 OK
IRQ 23 Intel(R) 82801EB USB2 Enhanced
Host Controller - 24DD OK
IRQ 13 Numeric data processor OK

IRQ 8 System CMOS/real time clock
OK
IRQ 0 System timer OK
IRQ 1 Standard 101/102-Key or Microsoft
Natural PS/2 Keyboard OK
IRQ 12 PS/2 Compatible Mouse OK

IRQ 4 Communications Port (COM1)
OK
IRQ 14 Primary IDE Channel OK

IRQ 15 Secondary IDE Channel OK

```

```

IRQ 10 Intel(R) 82801EB SMBus Controller
- 24D3 OK
[Memory]

Resource Device Status
0xA0000-0xBFFFF PCI bus OK

0xA0000-0xBFFFF RAGE XL PCI Family
(Microsoft Corporation) OK
0xD0000-0xD3FFF PCI bus OK

0xD4000-0xD7FFF PCI bus OK

0xD8000-0xDBFFF PCI bus OK

0x40000000-0xFEBFFFFF PCI bus OK

0xD8100000-0xD81FFFFF Intel(R) E7525/E7520
PCI Express Root Port B0 - 3597 OK

0xD8100000-0xD81FFFFF Broadcom NetXtreme
Gigabit Ethernet #3 OK
0xD8200000-0xD82FFFFF Intel(R) E7520 PCI
Express Root Port B1 - 3598 OK

0xD8200000-0xD82FFFFF Broadcom NetXtreme
Gigabit Ethernet #4 OK
0xD8300000-0xDBFFFFF Intel(R) E7520 PCI
Express Root Port C0 - 3599 OK

0xD8300000-0xDBFFFFF Intel(R)
6700/6702PXH I/OxAPIC Interrupt Controller A -
0326 OK
0xDA000000-0xDBFFFFF Intel(R) 6700PXH PCI
Express-to-PCI Bridge A - 0329 OK

0xDA000000-0xDBFFFFF Broadcom BCM5706C
NetXtreme II GigE OK
0xD8301000-0xD8301FFF Intel(R) 6700PXH
I/OxAPIC Interrupt Controller B - 0327 OK

0xD8001000-0xD80013FF Intel(R) 82801EB
USB2 Enhanced Host Controller - 24DD OK

0xDD000000-0xDDFFFFF RAGE XL PCI Family
(Microsoft Corporation) OK
0xDC000000-0xDC000FFF RAGE XL PCI Family
(Microsoft Corporation) OK
0xE0000000-0xEFFFFFFF Motherboard
resources OK

[Components]

[Multimedia]

[Audio Codecs]

CODEC Manufacturer Description Status
File Version Size
Creation Date
c:\windows\system32\msg723.acm
Microsoft Corporation
OK
C:\WINDOWS\system32\MSG723.A
CM 5.2.3790.1830 120.00
KB (122,880 bytes) 4/14/2005 10:01 AM

```

```

c:\windows\system32\msadp32.acm
Microsoft Corporation
OK
C:\WINDOWS\system32\MSADP32.
ACM 5.2.3790.0 (srv03_rtm.030324-
2048) 14.50 KB (14,848 bytes)
c:\windows\system32\msadp32.acm DSP
GROUP, INC. OK
C:\WINDOWS\system32\TSSOFT32.
ACM 1.01 9.50 KB (9,728 bytes)
4/12/2005 1:43 PM

c:\windows\system32\msgsm32.acm
Microsoft Corporation
OK
C:\WINDOWS\system32\MSGSM32.
ACM 5.2.3790.0 (srv03_rtm.030324-
2048) 20.50 KB (20,992 bytes)
4/12/2005 1:42 PM

c:\windows\system32\msg711.acm
Microsoft Corporation
OK
C:\WINDOWS\system32\MSG711.A
CM 5.2.3790.0 (srv03_rtm.030324-
2048) 10.00 KB (10,240 bytes)
4/12/2005 1:42 PM
c:\windows\system32\jmaadp32.acm
Microsoft Corporation
OK
C:\WINDOWS\system32\JMAADP32
.ACMT 5.2.3790.0 (srv03_rtm.030324-
2048) 15.50 KB (15,872 bytes)
4/12/2005 1:42 PM
c:\windows\system32\j3codeca.acm
Fraunhofer Institut Integrierte
Schaltungen IIS Fraunhofer IIS MPEG
Layer-3 Codec OK
C:\WINDOWS\system32\L3CODECA
.ACMT 1, 9, 0, 0305 284.00 KB (290,816
bytes) 4/12/2005 1:43 PM
c:\windows\system32\msaud32.acm
Microsoft Corporation Windows
Media Audio Codec OK
C:\WINDOWS\system32\MSAUD32.
ACM 8.00.00.4487 288.00 KB (294,912
bytes) 4/12/2005 1:43 PM
c:\windows\system32\sl_anet.acm Sipro
Lab Telecom Inc. Sipro Lab Telecom
Audio Codec OK
C:\WINDOWS\system32\SL_ANET.A
CM 3.02 84.00 KB (86,016
bytes) 4/12/2005 1:43 PM

[Video Codecs]

CODEC Manufacturer Description Status
File Version Size
Creation Date
c:\windows\system32\msh261.drv
Microsoft Corporation
OK
C:\WINDOWS\system32\MSH261.D
RV 5.2.3790.1830 184.00
KB (188,416 bytes) 4/14/2005 10:01 AM

c:\windows\system32\tsbyuv.dll
Microsoft Corporation
OK
C:\WINDOWS\system32\TSBYUV.D
LL 5.2.3790.0 (srv03_rtm.030324-
2048) 8.00 KB (8,192 bytes)
3/24/2003 5:50 PM

```



c:\windows\system32\msyuv.dll  
 Microsoft Corporation  
 OK  
 C:\WINDOWS\system32\MSYUV.DL  
 L 5.2.3790.0 (srv03\_rtm.030324-  
 2048) 16.50 KB (16,896 bytes)  
 3/24/2003 5:49 PM  
 c:\windows\system32\msvidc32.dll  
 Microsoft Corporation  
 OK  
 C:\WINDOWS\system32\MSVIDC32  
 .DLL 5.2.3790.0 (srv03\_rtm.030324-  
 2048) 26.50 KB (27,136 bytes)  
 4/12/2005 1:42 PM  
 c:\windows\system32\msrle32.dll  
 Microsoft Corporation  
 OK  
 C:\WINDOWS\system32\MSRLE32.  
 DLL 5.2.3790.0 (srv03\_rtm.030324-  
 2048) 10.50 KB (10,752 bytes)  
 4/12/2005 1:42 PM  
 c:\windows\system32\iyuv\_32.dll  
 Microsoft Corporation  
 OK  
 C:\WINDOWS\system32\IYUV\_32.D  
 LL 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 46.50 KB  
 (47,616 bytes) 4/14/2005 10:01 AM  
 c:\windows\system32\msh263.drv  
 Microsoft Corporation  
 OK  
 C:\WINDOWS\system32\MSH263.D  
 RV 5.2.3790.1830 288.00  
 KB (294,912 bytes) 4/14/2005 10:01 AM

[CD-ROM]

Item Value

[Sound Device]

Item Value

[Display]

Item Value  
 Name RAGE XL PCI Family (Microsoft  
 Corporation)  
 PNP Device ID  
 PCI\VEN\_1002&DEV\_4752&SUBSYS  
 \_3208103C&REV\_27\4&3A321F38&0&08F0

Adapter Type Not Available  
 Adapter Description ATI Technologies Inc.

Adapter RAM Not Available  
 Installed Drivers Not Available  
 Driver Version Not Available  
 INF File 5.10.3663.6013 (atiixpad.inf  
 section)  
 Color Planes ati2mpad  
 Color Table Entries Not Available  
 Resolution Not Available  
 Bits/Pixel Not Available  
 Memory Address 0xDD000000-  
 0xDDFFFFFF  
 I/O Port 0x00002000-0x000020FF  
 Memory Address 0xDC000000-  
 0xDC0000FF  
 IRQ Channel IRQ 16  
 I/O Port 0x000003B0-0x000003BB  
 I/O Port 0x000003C0-0x000003DF

Memory Address 0xA0000-0xBFFFF  
 Driver c:\windows\system32\drivers\ati2m  
 pad.sys (5.10.3663.6013, 335.38 KB (343,424  
 bytes), 4/13/2005 3:31 AM)

[Infrared]

Item Value

[Input]

[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&369939D9&0

Number of Function Keys 12  
 I/O Port 0x00000060-0x00000060  
 I/O Port 0x00000064-0x00000064  
 IRQ Channel IRQ 1  
 Driver

c:\windows\system32\drivers\i8042  
 prt.sys (5.2.3790.1830 (srv03\_sp1\_rtm.050324-  
 1447), 54.50 KB (55,808 bytes), 3/24/2003 5:01  
 PM)

[Pointing Device]

Item Value  
 Hardware Type PS/2 Compatible  
 Mouse  
 Number of Buttons 3  
 Status Error  
 PNP Device ID  
 ACPI\PNP0F13\4&369939D9&0

Power Management Supported No

Double Click Threshold 6  
 Handedness Right Handed Operation  
 IRQ Channel IRQ 12  
 Driver  
 c:\windows\system32\drivers\i8042  
 prt.sys (5.2.3790.1830 (srv03\_sp1\_rtm.050324-  
 1447), 54.50 KB (55,808 bytes), 3/24/2003 5:01  
 PM)

[Modem]

Item Value

[Network]

[Adapter]

Item Value  
 Name [00000001] RAS Async Adapter  
 Adapter Type Not Available  
 Product Type RAS Async Adapter  
 Installed Yes  
 PNP Device ID Not Available  
 Last Reset 7/7/2006 3:25 PM  
 Index 1

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 [00000002] WAN Miniport (L2TP)

Adapter Type Not Available  
 Product Type WAN Miniport (L2TP)  
 Installed Yes

PNP Device ID  
 ROOT\MS\_L2TPMINIPOINT\0000

Last Reset 7/7/2006 3:25 PM  
 Index 2

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\rasl2t  
 p.sys (5.2.3790.1830 (srv03\_sp1\_rtm.050324-  
 1447), 66.00 KB (67,584 bytes), 4/12/2005 1:42  
 PM)

Name [00000003] WAN Miniport (PPTP)

Adapter Type Wide Area Network (WAN)  
 Product Type WAN Miniport (PPTP)  
 Installed Yes  
 PNP Device ID  
 ROOT\MS\_PPTPMINIPOINT\0000

Last Reset 7/7/2006 3:25 PM  
 Index 3

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\raspp  
 tp.sys (5.2.3790.1830 (srv03\_sp1\_rtm.050324-  
 1447), 61.00 KB (62,464 bytes), 4/12/2005 1:42  
 PM)

Name [00000004] WAN Miniport (PPPOE)

Adapter Type Wide Area Network (WAN)  
 Product Type WAN Miniport (PPPOE)  
 Installed Yes  
 PNP Device ID  
 ROOT\MS\_PPPOEMINIPOINT\0000

Last Reset 7/7/2006 3:25 PM  
 Index 4

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\rasppoe.sys (5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447), 40.00 KB (40,960 bytes), 4/12/2005 1:42 PM)  
 Name [00000005] Direct Parallel  
 Adapter TypeNot Available  
 Product TypeDirect Parallel  
 Installed Yes  
 PNP Device ID  
 ROOT\MS\_PTIMINIPORT\0000  
 Last Reset 7/7/2006 3:25 PM  
 Index 5  
 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 (5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447), 19.50 KB (19,968 bytes), 4/12/2005 1:42 PM)  
 Name [00000006] WAN Miniport (IP)  
 Adapter TypeNot Available  
 Product TypeWAN Miniport (IP)  
 Installed Yes  
 PNP Device ID  
 ROOT\MS\_NDISWANIP\0000  
 Last Reset 7/7/2006 3:25 PM  
 Index 6  
 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 (5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447), 91.00 KB (93,184 bytes), 4/12/2005 1:42 PM)  
 Name [00000007] Broadcom NetXtreme Gigabit Ethernet  
 Adapter TypeNot Available  
 Product TypeBroadcom NetXtreme Gigabit Ethernet  
 Installed Yes  
 PNP Device ID  
 Last Reset 7/7/2006 3:25 PM  
 Index 7  
 Service Name b57w2k  
 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**  
 MAC Address Not Available  
 Name [00000008] Broadcom NetXtreme Gigabit Ethernet  
 Adapter TypeNot Available  
 Product TypeBroadcom NetXtreme Gigabit Ethernet  
 Installed Yes  
 PNP Device ID  
 Last Reset 7/7/2006 3:25 PM  
 Index 8  
 Service Name b57w2k  
 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] WAN Miniport (Network Monitor)  
 Adapter TypeNot Available  
 Product TypeWAN Miniport (Network Monitor)  
 Installed Yes  
 PNP Device ID  
 ROOT\MS\_NDISWANBH\0000  
 Last Reset 7/7/2006 3:25 PM  
 Index 9  
 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 (5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447), 91.00 KB (93,184 bytes), 4/12/2005 1:42 PM)  
 Name [00000010] Broadcom NetXtreme Gigabit Ethernet  
 Adapter TypeEthernet 802.3  
 Product TypeBroadcom NetXtreme Gigabit Ethernet  
 Installed Yes  
 PNP Device ID  
 PCI\VEN\_14E4&DEV\_1659&SUBSYS\_1659103C&REV\_11\4&253DB27D&0&0020  
 Last Reset 7/7/2006 3:25 PM  
 Index 10  
 Service Name b57w2k  
 IP Address 15.1.102.18  
 IP Subnet 255.255.0.0  
 Default IP Gateway Not Available  
 DHCP Enabled No  
 DHCP Server Not Available  
 DHCP Lease Expires Not Available  
 DHCP Lease Obtained Not Available  
 MAC Address 00:15:60:09:E1:18  
 Memory Address 0xD8100000-0xD81FFFFF  
 IRQ Channel IRQ 16

Driver  
 c:\windows\system32\drivers\b57xp32.sys (7.86.0.0 built by: WinDDK, 118.63 KB (121,472 bytes), 4/13/2005 4:24 PM)  
 Name [00000011] Broadcom NetXtreme Gigabit Ethernet  
 Adapter TypeEthernet 802.3  
 Product TypeBroadcom NetXtreme Gigabit Ethernet  
 Installed Yes  
 PNP Device ID  
 PCI\VEN\_14E4&DEV\_1659&SUBSYS\_1659103C&REV\_11\4&1C834E48&0&0028  
 Last Reset 7/7/2006 3:25 PM  
 Index 11  
 Service Name b57w2k  
 IP Address 15.1.103.18  
 IP Subnet 255.255.0.0  
 Default IP Gateway Not Available  
 DHCP Enabled No  
 DHCP Server Not Available  
 DHCP Lease Expires Not Available  
 DHCP Lease Obtained Not Available  
 MAC Address 00:15:60:09:E1:19  
 Memory Address 0xD8200000-0xD82FFFFF  
 IRQ Channel IRQ 16  
 Driver  
 c:\windows\system32\drivers\b57xp32.sys (7.86.0.0 built by: WinDDK, 118.63 KB (121,472 bytes), 4/13/2005 4:24 PM)  
 [Protocol]  
 Item Value  
 Name MSAFD Tcpi [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 MSAFD Tcpi [UDP/IP]  
 Connectionless Service Yes  
 Guarantees Delivery No  
 Guarantees Sequencing No  
 Maximum Address Size 16 bytes  
 Maximum Message Size 63.93 KB (65,467 bytes)  
 Message Oriented Yes  
 Minimum Address Size 16 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting Yes  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption No  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting Yes

Name RSVP UDP Service Provider  
 Connectionless Service Yes  
 Guarantees Delivery No  
 Guarantees Sequencing No  
 Maximum Address Size 16 bytes  
 Maximum Message Size 63.93 KB (65,467 bytes)  
 Message Oriented Yes  
 Minimum Address Size 16 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting Yes  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption Yes  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting Yes

Name RSVP TCP Service Provider  
 Connectionless Service No  
 Guarantees Delivery Yes  
 Guarantees Sequencing Yes  
 Maximum Address Size 16 bytes  
 Maximum Message Size 0 bytes  
 Message Oriented No  
 Minimum Address Size 16 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting No  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption Yes  
 Supports Expedited Data Yes  
 Supports Graceful Closing Yes  
 Supports Guaranteed Bandwidth No

Supports Multicasting No

Name MSAFD NetBIOS  
 [\Device\NetBT\_Tcpip\_{5522ED49-FD5C-4788-B78F-57A4F86E5748}] SEQPACKET 5  
 Connectionless Service No  
 Guarantees Delivery Yes  
 Guarantees Sequencing Yes  
 Maximum Address Size 20 bytes  
 Maximum Message Size 62.50 KB (64,000 bytes)  
 Message Oriented Yes  
 Minimum Address Size 20 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting No  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption No  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting No

Name MSAFD NetBIOS  
 [\Device\NetBT\_Tcpip\_{5522ED49-FD5C-4788-B78F-57A4F86E5748}] DATAGRAM 5  
 Connectionless Service Yes  
 Guarantees Delivery No  
 Guarantees Sequencing No  
 Maximum Address Size 20 bytes  
 Maximum Message Size 62.50 KB (64,000 bytes)  
 Message Oriented Yes  
 Minimum Address Size 20 bytes  
 Pseudo Stream Oriented No

Supports Broadcasting No

Supports Disconnect Data No  
 Supports Encryption No  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting No

Name MSAFD NetBIOS  
 [\Device\NetBT\_Tcpip\_{F51C171B-E007-42CE-AD2F-A3FD3FFC2C99}] SEQPACKET 4  
 Connectionless Service No  
 Guarantees Delivery Yes  
 Guarantees Sequencing Yes  
 Maximum Address Size 20 bytes  
 Maximum Message Size 62.50 KB (64,000 bytes)  
 Message Oriented Yes  
 Minimum Address Size 20 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting No  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption No  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting No

Name MSAFD NetBIOS  
 [\Device\NetBT\_Tcpip\_{F51C171B-E007-42CE-AD2F-A3FD3FFC2C99}] DATAGRAM 4  
 Connectionless Service Yes  
 Guarantees Delivery No  
 Guarantees Sequencing No  
 Maximum Address Size 20 bytes  
 Maximum Message Size 62.50 KB (64,000 bytes)  
 Message Oriented Yes  
 Minimum Address Size 20 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting Yes  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption No  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting No

Name MSAFD NetBIOS  
 [\Device\NetBT\_Tcpip\_{356451BD-E9BA-4CDE-ADFA-95D45FC47A55}] SEQPACKET 3  
 Connectionless Service No  
 Guarantees Delivery Yes  
 Guarantees Sequencing Yes  
 Maximum Address Size 20 bytes  
 Maximum Message Size 62.50 KB (64,000 bytes)  
 Message Oriented Yes  
 Minimum Address Size 20 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting No  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption No  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting No

Name MSAFD NetBIOS  
 [\Device\NetBT\_Tcpip\_{356451BD-E9BA-4CDE-ADFA-95D45FC47A55}] DATAGRAM 3  
 Connectionless Service No  
 Guarantees Delivery Yes  
 Guarantees Sequencing Yes  
 Maximum Address Size 20 bytes  
 Maximum Message Size 62.50 KB (64,000 bytes)  
 Message Oriented Yes  
 Minimum Address Size 20 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting No  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption No  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting No

Name MSAFD NetBIOS  
 [\Device\NetBT\_Tcpip\_{356451BD-E9BA-4CDE-ADFA-95D45FC47A55}] DATAGRAM 3  
 Connectionless Service No  
 Guarantees Delivery No  
 Guarantees Sequencing No  
 Maximum Address Size 20 bytes  
 Maximum Message Size 62.50 KB (64,000 bytes)  
 Message Oriented Yes  
 Minimum Address Size 20 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting Yes  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption No  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting No

Name MSAFD NetBIOS  
 [\Device\NetBT\_Tcpip\_{6C8EF2B0-C570-41AD-BA17-47DDB867B581}] SEQPACKET 0  
 Connectionless Service No  
 Guarantees Delivery Yes  
 Guarantees Sequencing Yes  
 Maximum Address Size 20 bytes  
 Maximum Message Size 62.50 KB (64,000 bytes)  
 Message Oriented Yes  
 Minimum Address Size 20 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting No  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption No  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting No

Name MSAFD NetBIOS  
 [\Device\NetBT\_Tcpip\_{6C8EF2B0-C570-41AD-BA17-47DDB867B581}] DATAGRAM 0  
 Connectionless Service Yes  
 Guarantees Delivery No  
 Guarantees Sequencing No  
 Maximum Address Size 20 bytes  
 Maximum Message Size 62.50 KB (64,000 bytes)  
 Message Oriented Yes  
 Minimum Address Size 20 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting Yes  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption No  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting No

Name MSAFD NetBIOS  
 [\Device\NetBT\_Tcpip\_{C09F1971-DD34-4CF7-86C3-79FC40F767E0}] SEQPACKET 1  
 Connectionless Service No  
 Guarantees Delivery Yes  
 Guarantees Sequencing Yes  
 Maximum Address Size 20 bytes

Maximum Message Size 62.50 KB (64,000 bytes)  
 Message Oriented Yes  
 Minimum Address Size 20 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting No  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption No  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting No

Name MSAFD NetBIOS  
 [\Device\NetBT\_Tcpip\_{C09F1971-DD34-4CF7-86C3-79FC40F767E0}] DATAGRAM 1  
 Connectionless Service Yes  
 Guarantees Delivery No  
 Guarantees Sequencing No  
 Maximum Address Size 20 bytes  
 Maximum Message Size 62.50 KB (64,000 bytes)  
 Message Oriented Yes  
 Minimum Address Size 20 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting Yes  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption No  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting No

Name MSAFD NetBIOS  
 [\Device\NetBT\_Tcpip\_{F406FAC2-3DB8-4D61-8565-B2072EECB1F7}] SEQUENCE 2  
 Connectionless Service No  
 Guarantees Delivery Yes  
 Guarantees Sequencing Yes  
 Maximum Address Size 20 bytes  
 Maximum Message Size 62.50 KB (64,000 bytes)  
 Message Oriented Yes  
 Minimum Address Size 20 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting No  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption No  
 Supports Expedited Data No  
 Supports Graceful Closing No  
 Supports Guaranteed Bandwidth No

Supports Multicasting No

Name MSAFD NetBIOS  
 [\Device\NetBT\_Tcpip\_{F406FAC2-3DB8-4D61-8565-B2072EECB1F7}] DATAGRAM 2  
 Connectionless Service Yes  
 Guarantees Delivery No  
 Guarantees Sequencing No  
 Maximum Address Size 20 bytes  
 Maximum Message Size 62.50 KB (64,000 bytes)  
 Message Oriented Yes  
 Minimum Address Size 20 bytes  
 Pseudo Stream Oriented No  
 Supports Broadcasting Yes  
 Supports Connect Data No  
 Supports Disconnect Data No  
 Supports Encryption No

Supports Expedited Data No  
 Supports Guaranteed Bandwidth No

Supports Multicasting No

[WinSock]

Item Value  
 File c:\windows\system32\winsock.dll  
 Size 2.80 KB (2,864 bytes)  
 Version 3.10

File c:\windows\system32\wssock32.dll

Size 22.00 KB (22,528 bytes)  
 Version 5.2.3790.0 (srv03\_rtm.030324-2048)

[Ports]

[Serial]

Item Value  
 Name Communications Port (COM1)

Status OK  
 PNP Device ID ACPI\PNP0501\1

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 0  
 XOffXMit Threshold 0  
 XOn Character 0  
 XOnXMit Threshold 0  
 XOnXOff InFlow Control 0  
 XOnXOff OutFlow Control 0  
 I/O Port 0x000003F8-0x000003FF  
 IRQ Channel IRQ 4

Driver  
 c:\windows\system32\drivers\serial.sys (5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447), 64.00 KB (65,536 bytes), 3/24/2003 3:40 PM)  
 [Parallel]

Item Value

[Storage]

[Drives]

Item Value  
 Drive C:  
 Description Local Fixed Disk  
 Compressed No  
 File System NTFS  
 Size 74.53 GB (80,023,715,840 bytes)

Free Space 69.14 GB (74,237,181,952 bytes)

Volume Name

Volume Serial Number DCCFCBF1

[Disks]

Item Value  
 Description Disk drive  
 Manufacturer (Standard disk drives)  
 Model Maxtor 6L080M0  
 Bytes/Sector 512  
 Media Loaded Yes  
 Media Type Fixed hard disk  
 Partitions 1  
 SCSI Bus 0  
 SCSI Logical Unit 0  
 SCSI Port 0  
 SCSI Target ID 0  
 Sectors/Track 63  
 Size 74.53 GB (80,023,749,120 bytes)

Total Cylinders 9,729  
 Total Sectors 156,296,385  
 Total Tracks 2,480,895  
 Tracks/Cylinder 255  
 Partition Disk #0, Partition #0  
 Partition Size 74.53 GB (80,023,716,864 bytes)

Partition Starting Offset 32,256 bytes

[SCSI]

Item Value

[IDE]

Item Value  
 Name Intel(R) 82801EB Ultra ATA Storage  
 Controllers  
 Manufacturer Intel  
 Status OK  
 PNP Device ID  
 PCI\VEN\_8086&DEV\_24D1&SUBSYS\_3208103C&REV\_02\3&61AAA01&0&FA  
 I/O Port 0x00001470-0x0000147F  
 Driver  
 c:\windows\system32\drivers\pciide.sys (5.2.3790.0 (srv03\_rtm.030324-2048), 5.50 KB (5,632 bytes), 3/24/2003 3:04 PM)

Name Primary IDE Channel  
 Manufacturer (Standard IDE ATA/ATAPI controllers)  
 Status OK  
 PNP Device ID  
 PCI\IDE\IDECHANNEL\4&1D65F1F&0&0  
 I/O Port 0x000001F0-0x000001F7  
 I/O Port 0x000003F6-0x000003F6  
 IRQ Channel IRQ 14  
 Driver  
 c:\windows\system32\drivers\atapi.sys (5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447), 93.50 KB (95,744 bytes), 3/24/2003 3:04 PM)

Name Secondary IDE Channel  
 Manufacturer (Standard IDE ATA/ATAPI controllers)  
 Status OK  
 PNP Device ID  
 PCI\IDE\IDECHANNEL\4&1D65F1F&0&1  
 I/O Port 0x00000170-0x00000177  
 I/O Port 0x00000376-0x00000376  
 IRQ Channel IRQ 15  
 Driver  
 c:\windows\system32\drivers\atapi.sys (5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447), 93.50 KB (95,744 bytes), 3/24/2003 3:04 PM)

[Printing]

| Name | Driver | Port Name | Server Name |
|------|--------|-----------|-------------|
|      |        |           |             |

[Problem Devices]

| Device Code   | PNP Device ID             | Error  |
|---|---------------------------|--|
| Standard 101/102-Key or Microsoft Natural PS/2 Keyboard | ACPI\PNP0303\4&369939D9&0 | This device is not present, is not working properly, or does not have all its drivers installed. |
| PS/2 Compatible Mouse                                   | ACPI\PNP0F13\4&369939D9&0 | This device is not present, is not working properly, or does not have all its drivers installed. |

[USB]

| Device  | PNP Device ID   |
|---|---|
| Intel(R) 82801EB USB Universal Host Controller - 24D2 | PCI\VEN_8086&DEV_24D2&SUBSYS_3208103C&REV_02\3&61AAA01&0&E8 |
| Intel(R) 82801EB USB Universal Host Controller - 24D4 | PCI\VEN_8086&DEV_24D4&SUBSYS_3208103C&REV_02\3&61AAA01&0&E9 |
| Intel(R) 82801EB USB2 Enhanced Host Controller - 24DD | PCI\VEN_8086&DEV_24DD&SUBSYS_3208103C&REV_02\3&61AAA01&0&EF |

[Software Environment]

[System Drivers]

| Name     | Description                            | File                                     | Type          | Started | Start Mode    | State         |
|----------|--|--|---------------|---------|---------------|---------------|
| pluseisk | Accepts                                | Not Available                            | Kernel        | Stopped | Not Available | Stopped       |
| pluseisk | Driver                                 | OK                                       | Ignore        | No      | Normal        | No            |
| acpi     | Microsoft ACPI Driver                  | c:\windows\system32\drivers\acpi.sys     | Kernel Driver | Running | OK            | Normal        |
| ys       | Kernel Driver                          | Yes                                      | Boot          | Running | OK            | Normal        |
| acpiec   | ACPIEC                                 | c:\windows\system32\drivers\acpiec.sys   | Kernel Driver | Stopped | OK            | Disabled      |
| adpu160m | adpu160m                               | Not Available                            | Kernel        | Stopped | Disabled      | Stopped       |
| adpu160m | Driver                                 | OK                                       | Normal        | No      | Normal        | No            |
| adpu320  | adpu320                                | Not Available                            | Kernel        | Stopped | Disabled      | Stopped       |
| adpu320  | Driver                                 | OK                                       | Normal        | No      | Normal        | No            |
| afcnt    | afcnt                                  | Not Available                            | Kernel        | Stopped | Disabled      | Stopped       |
| afcnt    | Driver                                 | OK                                       | Normal        | No      | Normal        | No            |
| afd      | AFD Networking Support Environment     | c:\windows\system32\drivers\afd.sys      | Kernel Driver | Running | OK            | System Normal |
| aha154x  | Aha154x                                | Not Available                            | Kernel        | Stopped | Disabled      | Stopped       |
| aha154x  | Driver                                 | OK                                       | Normal        | No      | Normal        | No            |
| aic78u2  | aic78u2                                | Not Available                            | Kernel        | Stopped | Disabled      | Stopped       |
| aic78u2  | Driver                                 | OK                                       | Normal        | No      | Normal        | No            |
| aic78xx  | aic78xx                                | Not Available                            | Kernel        | Stopped | Disabled      | Stopped       |
| aic78xx  | Driver                                 | OK                                       | Normal        | No      | Normal        | No            |
| aliide   | AliIde                                 | Not Available                            | Kernel        | Stopped | Disabled      | Stopped       |
| aliide   | Driver                                 | OK                                       | Normal        | No      | Normal        | No            |
| alkernel | Altiris Kernel Driver                  | c:\windows\system32\drivers\alkernel.sys | Kernel Driver | Running | OK            | Manual Normal |
| asynmac  | RAS Asynchronous Media Driver          | c:\windows\system32\drivers\asynmac.sys  | Kernel Driver | Stopped | OK            | Manual Normal |
| atapi    | Standard IDE/ESDI Hard Disk Controller | c:\windows\system32\drivers\atapi.sys    | Kernel Driver | Running | OK            | Boot Normal   |
| atdisk   | Atdisk                                 | Not Available                            | Kernel        | Stopped | Disabled      | Stopped       |
| atdisk   | Driver                                 | OK                                       | Ignore        | No      | Normal        | No            |

|           |                                     |  |                    |          |          |                 |
|-----------|-------------------------------------|--|--------------------|----------|----------|-----------------|
| ati2mpad  | ati2mpad                            | c:\windows\system32\drivers\ati2mpad.sys | Kernel Driver      | Running  | OK       | Manual Ignore   |
| atmarpc   | ATM ARP Client Protocol             | c:\windows\system32\drivers\atmarpc.sys  | Kernel Driver      | Stopped  | OK       | Manual Normal   |
| audstub   | Audio Stub Driver                   | c:\windows\system32\drivers\audstub.sys  | Kernel Driver      | Running  | OK       | Manual Normal   |
| b06bdrv   | Broadcom NetXtreme II VBD           | c:\windows\system32\drivers\b06bdrv.sys  | Kernel Driver      | Running  | OK       | Manual Normal   |
| b57w2k    | Broadcom NetXtreme Gigabit Ethernet | c:\windows\system32\drivers\b57w2k.sys   | Kernel Driver      | Running  | OK       | Manual Normal   |
| beep      | Beep                                | c:\windows\system32\drivers\beep.sys     | Kernel Driver      | Running  | OK       | System Normal   |
| cbidf2k   | cbidf2k                             | c:\windows\system32\drivers\cbidf2k.sys  | Kernel Driver      | Stopped  | OK       | Disabled Normal |
| cd20xrnt  | cd20xrnt                            | Not Available                            | Kernel             | Stopped  | Disabled | Stopped         |
| cd20xrnt  | Driver                              | OK                                       | Normal             | No       | Normal   | No              |
| cdfs      | Cdfs                                | c:\windows\system32\drivers\cdfs.sys     | File System Driver | Disabled | Stopped  | OK              |
| cdrom     | Cdrom                               | c:\windows\system32\drivers\cdrom.sys    | Kernel Driver      | Stopped  | OK       | System Ignore   |
| changer   | Changer                             | Not Available                            | Kernel             | Stopped  | Disabled | Stopped         |
| changer   | Driver                              | OK                                       | Ignore             | No       | Normal   | No              |
| clusdisk  | Cluster Disk Driver                 | c:\windows\system32\drivers\clusdisk.sys | Kernel Driver      | Stopped  | OK       | Disabled Normal |
| cmdide    | CmdIde                              | Not Available                            | Kernel             | Stopped  | Disabled | Stopped         |
| cmdide    | Driver                              | OK                                       | Normal             | No       | Normal   | No              |
| cpqarray  | Cpqarray                            | Not Available                            | Kernel             | Stopped  | Disabled | Stopped         |
| cpqarray  | Driver                              | OK                                       | Normal             | No       | Normal   | No              |
| cpqarray2 | Cpqarray2                           | Not Available                            | Kernel             | Stopped  | Disabled | Stopped         |
| cpqarray2 | Driver                              | OK                                       | Normal             | No       | Normal   | No              |

|                    |  |                                     |                         |
|--------------------|--|-------------------------------------|-------------------------|
| cpqccism<br>Driver | cpqccism<br>No<br>OK   | Not Available<br>Disabled<br>Normal | Kernel<br>Stopped<br>No |
| cpqfcalm<br>Driver | cpqfcalm<br>No<br>OK   | Not Available<br>Disabled<br>Normal | Kernel<br>Stopped<br>No |
| crdisk             | CRC Disk Filter Driver   |                                     |                         |
| k.sys              | c:\windows\system32\drivers\crdis<br>Kernel Driver<br>Running        | Yes<br>OK                           | Boot<br>Normal          |
| dac960nt<br>Driver | dac960nt<br>No<br>OK   | Not Available<br>Disabled<br>Normal | Kernel<br>Stopped<br>No |
| dellcerc<br>Driver | dellcerc<br>No<br>OK   | Not Available<br>Disabled<br>Normal | Kernel<br>Stopped<br>No |
| dfsdriver          | DfsDriver  |                                     |                         |
| s                  | c:\windows\system32\drivers\dfs.sy<br>File System Driver<br>Boot     | Yes<br>Running                      | OK<br>OK                |
| disk               | Disk Driver  |                                     |                         |
| ys                 | c:\windows\system32\drivers\disk.s<br>Kernel Driver<br>Running       | Yes<br>OK                           | Boot<br>Normal          |
| dmbboot            | dmbboot  |                                     |                         |
| ot.sys             | c:\windows\system32\drivers\dmbo<br>Kernel Driver<br>Stopped         | No<br>OK                            | Disabled<br>Normal      |
| dmio               | Logical Disk Manager Driver  |                                     |                         |
| sys                | c:\windows\system32\drivers\dmio.<br>Kernel Driver<br>Running        | Yes<br>OK                           | Boot<br>Normal          |
| dmload             | dmload   |                                     |                         |
| ad.sys             | c:\windows\system32\drivers\dmlo<br>Kernel Driver<br>Running         | Yes<br>OK                           | Boot<br>Normal          |
| dpti2o<br>Driver   | dpti2o<br>No<br>OK   | Not Available<br>Disabled<br>Normal | Kernel<br>Stopped<br>No |
| fastfat            | Fastfat  |                                     |                         |
| t.sys              | c:\windows\system32\drivers\fastfa<br>File System Driver<br>Disabled | No<br>Stopped                       | OK<br>OK                |
| fdc                | Fdc  |                                     |                         |
| s                  | c:\windows\system32\drivers\fdc.sy<br>Kernel Driver<br>Stopped       | No<br>OK                            | System<br>Ignore        |
| fips               | Fips   |                                     |                         |
| ys                 | c:\windows\system32\drivers\fips.s<br>Kernel Driver<br>Running       | Yes<br>OK                           | System<br>Normal        |
| flpydisk           | Flpydisk   |                                     |                         |
| sk.sys             | c:\windows\system32\drivers\flpydi<br>Kernel Driver<br>Stopped       | No<br>OK                            | System<br>Ignore        |

|                         |  |                                     |                         |
|-------------------------|--|-------------------------------------|-------------------------|
| fltmgr                  | FltMgr   |                                     |                         |
| .sys                    | c:\windows\system32\drivers\fltmgr<br>File System Driver<br>Boot | Yes<br>Running                      | OK<br>OK                |
| ftdisk                  | Volume Manager Driver  |                                     |                         |
| sys                     | c:\windows\system32\drivers\ftdisk.<br>Kernel Driver<br>Running  | Yes<br>OK                           | Boot<br>Normal          |
| gpc                     | Generic Packet Classifier  |                                     |                         |
| c.sys                   | c:\windows\system32\drivers\msgp<br>Kernel Driver<br>Running     | Yes<br>OK                           | Manual<br>Normal        |
| hpn<br>Driver           | hpn<br>No<br>OK  | Not Available<br>Disabled<br>Normal | Kernel<br>Stopped<br>No |
| hpt3xx<br>Driver        | hpt3xx<br>No<br>OK   | Not Available<br>Disabled<br>Normal | Kernel<br>Stopped<br>No |
| http                    | HTTP   |                                     |                         |
| ys                      | c:\windows\system32\drivers\http.s<br>Kernel Driver<br>Running   | Yes<br>OK                           | Manual<br>Normal        |
| i2omgmt<br>Driver       | i2omgmt<br>No<br>OK  | Not Available<br>System<br>Normal   | Kernel<br>Stopped<br>No |
| i2omp<br>Driver         | i2omp<br>No<br>OK  | Not Available<br>Disabled<br>Normal | Kernel<br>Stopped<br>No |
| i8042prt<br>Port Driver | i8042 Keyboard and PS/2 Mouse                                    |                                     |                         |
| prt.sys                 | c:\windows\system32\drivers\i8042<br>Kernel Driver<br>Running    | Yes<br>OK                           | System<br>Normal        |
| iirsp<br>Driver         | iirsp<br>No<br>OK  | Not Available<br>Disabled<br>Normal | Kernel<br>Stopped<br>No |
| intelide                | IntelIde   |                                     |                         |
| e.sys                   | c:\windows\system32\drivers\intellid<br>Kernel Driver<br>Running | Yes<br>OK                           | Boot<br>Normal          |
| intelppm                | Intel Processor Driver   |                                     |                         |
| pm.sys                  | c:\windows\system32\drivers\intelp<br>Kernel Driver<br>Running   | Yes<br>OK                           | Manual<br>Normal        |
| ip6fw                   | IPv6 Windows Firewall Driver                                     |                                     |                         |
| .sys                    | c:\windows\system32\drivers\ip6fw<br>Kernel Driver<br>Stopped    | No<br>OK                            | Manual<br>Normal        |
| ipfilterdriver          | IP Traffic Filter Driver   |                                     |                         |
| v.sys                   | c:\windows\system32\drivers\ipftdr<br>Kernel Driver<br>Stopped   | No<br>OK                            | Manual<br>Normal        |
| ipinip                  | IP in IP Tunnel Driver   |                                     |                         |
| sys                     | c:\windows\system32\drivers\ipinip.<br>Kernel Driver<br>Stopped  | No<br>OK                            | Manual<br>Normal        |

|                    |  |                                     |                         |
|--------------------|--|-------------------------------------|-------------------------|
| ipnat              | IP Network Address Translator                                    |                                     |                         |
| sys                | c:\windows\system32\drivers\ipnat.<br>Kernel Driver<br>Stopped   | No<br>OK                            | Manual<br>Normal        |
| ipsec              | IPSEC driver   |                                     |                         |
| sys                | c:\windows\system32\drivers\ipsec.<br>Kernel Driver<br>Running   | Yes<br>OK                           | System<br>Normal        |
| ipsraidn<br>Driver | ipsraidn<br>No<br>OK   | Not Available<br>Disabled<br>Normal | Kernel<br>Stopped<br>No |
| irenum             | IR Enumerator Service  |                                     |                         |
| m.sys              | c:\windows\system32\drivers\irenu<br>Kernel Driver<br>Stopped    | No<br>OK                            | Manual<br>Normal        |
| isapnp             | PnP ISA/EISA Bus Driver  |                                     |                         |
| p.sys              | c:\windows\system32\drivers\isapn<br>Kernel Driver<br>Running    | Yes<br>OK                           | Boot<br>Critical        |
| kbdclass           | Keyboard Class Driver  |                                     |                         |
| ass.sys            | c:\windows\system32\drivers\kbdc<br>Kernel Driver<br>Running     | Yes<br>OK                           | System<br>Normal        |
| ksecdd             | KSecDD   |                                     |                         |
| d.sys              | c:\windows\system32\drivers\ksecd<br>Kernel Driver<br>Running    | Yes<br>OK                           | Boot<br>Normal          |
| l2nd               | Broadcom NetXtreme II BXND                                       |                                     |                         |
| 51x.sys            | c:\windows\system32\drivers\bxnd<br>Kernel Driver<br>Stopped     | No<br>OK                            | Manual<br>Normal        |
| lp6nds35<br>Driver | lp6nds35<br>No<br>OK   | Not Available<br>Disabled<br>Normal | Kernel<br>Stopped<br>No |
| mnmdd              | mnmdd  |                                     |                         |
| dd.sys             | c:\windows\system32\drivers\mnm<br>Kernel Driver<br>Running      | Yes<br>OK                           | System<br>Ignore        |
| modem              | Modem  |                                     |                         |
| m.sys              | c:\windows\system32\drivers\mode<br>Kernel Driver<br>Stopped     | No<br>OK                            | Manual<br>Ignore        |
| mouclass           | Mouse Class Driver   |                                     |                         |
| ass.sys            | c:\windows\system32\drivers\moucl<br>Kernel Driver<br>Running    | Yes<br>OK                           | System<br>Normal        |
| mountmgr           | Mount Point Manager  |                                     |                         |
| tmgr.sys           | c:\windows\system32\drivers\moun<br>Kernel Driver<br>Running     | Yes<br>OK                           | Boot<br>Normal          |
| mraid35x<br>Driver | mraid35x<br>No<br>OK   | Not Available<br>Disabled<br>Normal | Kernel<br>Stopped<br>No |
| mrxdav             | WebDav Client Redirector   |                                     |                         |
| av.sys             | c:\windows\system32\drivers\mrxd<br>File System Driver<br>Manual | No<br>Stopped                       | OK<br>OK                |

|          |  |
|----------|--|
| mrxsmb   | MRXSMB   |
| mb.sys   | File System Driver Yes<br>System Running OK<br>Normal No Yes                                       |
| msfs     | Msf  |
| sys      | c:\windows\system32\drivers\msfs.<br>File System Driver Yes<br>System Running OK<br>Normal No Yes  |
| mssmbios | Microsoft System Management<br>BIOS Driver   |
| bios.sys | c:\windows\system32\drivers\mssm<br>Kernel Driver Yes Manual<br>Running OK Normal<br>No Yes        |
| mup      | Mup  |
| sys      | c:\windows\system32\drivers\mup.<br>File System Driver Yes<br>Boot Running OK<br>Normal No Yes     |
| ndis     | NDIS System Driver   |
| ys       | c:\windows\system32\drivers\ndis.s<br>Kernel Driver Yes Boot<br>Running OK Normal<br>No Yes        |
| ndistapi | Remote Access NDIS TAPI Driver   |
| pi.sys   | c:\windows\system32\drivers\ndista<br>Kernel Driver Yes Manual<br>Running OK Normal<br>No Yes      |
| ndisui   | NDIS Usermode I/O Protocol   |
| o.sys    | c:\windows\system32\drivers\ndisui<br>Kernel Driver Yes Manual<br>Running OK Normal<br>No Yes      |
| ndiswan  | Remote Access NDIS WAN Driver  |
| an.sys   | c:\windows\system32\drivers\ndisw<br>Kernel Driver Yes Manual<br>Running OK Normal<br>No Yes       |
| ndproxy  | NDIS Proxy   |
| xy.sys   | c:\windows\system32\drivers\ndpro<br>Kernel Driver Yes Manual<br>Running OK Normal<br>No Yes       |
| netbios  | NetBIOS Interface  |
| os.sys   | c:\windows\system32\drivers\netbi<br>File System Driver Yes<br>System Running OK<br>Normal No Yes  |
| netbt    | NetBios over Tcpip   |
| sys      | c:\windows\system32\drivers\netbt.<br>Kernel Driver Yes System<br>Running OK Normal<br>No Yes      |
| nfrd960  | nfrd960 Not Available Kernel<br>Driver No Disabled Stopped<br>OK Normal No                         |
| nm       | Network Monitor Driver   |
| sys      | c:\windows\system32\drivers\nmnt.<br>Kernel Driver No Manual<br>Stopped OK Normal<br>No No         |
| npfs     | Npfs   |
| ys       | c:\windows\system32\drivers\npfs.s<br>File System Driver Yes<br>System Running OK<br>Normal No Yes |

|              |   |
|--------------|---|
| ntfs         | Ntfs  |
| ys           | c:\windows\system32\drivers\ntfs.s<br>File System Driver Yes<br>Disabled Running OK<br>Normal No Yes                  |
| null         | Null  |
| ys           | c:\windows\system32\drivers\null.s<br>Kernel Driver Yes System<br>Running OK Normal<br>No Yes                         |
| parport      | Parport   |
| rt.sys       | c:\windows\system32\drivers\parpo<br>Kernel Driver No Manual<br>Stopped OK Ignore<br>No No                            |
| partmgr      | Partition Manager   |
| gr.sys       | c:\windows\system32\drivers\partm<br>Kernel Driver Yes Boot<br>Running OK Normal<br>No Yes                            |
| pci          | PCI Bus Driver  |
| s            | c:\windows\system32\drivers\pci.sy<br>Kernel Driver Yes Boot<br>Running OK Critical<br>No Yes                         |
| pciide       | PCIIde  |
| .sys         | c:\windows\system32\drivers\pciide<br>Kernel Driver Yes Boot<br>Running OK Normal<br>No Yes                           |
| pcmcia       | Pcmcia  |
| a.sys        | c:\windows\system32\drivers\pcmc<br>Kernel Driver No Disabled<br>Stopped OK Normal<br>No No                           |
| pdcomp       | PDCOMP Not Available Kernel<br>Driver No Manual Stopped<br>OK OK Ignore No  |
| pdframe      | PDFRAME Not Available Kernel<br>Driver No Manual Stopped<br>OK OK Ignore No   |
| pdreli       | PDRELI Not Available Kernel<br>Driver No Manual Stopped<br>OK OK Ignore No  |
| pdframe      | PDRFRAME Not Available Kernel<br>Driver No Manual Stopped<br>OK OK Ignore No  |
| perc2        | perc2 Not Available Kernel<br>Driver No Disabled Stopped<br>OK OK Normal No   |
| perc2hib     | perc2hib Not Available Kernel<br>Driver No Disabled Stopped<br>OK OK Normal No  |
| pgtrackr01   | Page Tracker1 for X86perfsys<br>\\?\c:\top1.1\pgtrackr01.sys<br>Kernel Driver No Manual<br>Stopped OK Ignore<br>No No |
| pptpminiport | WAN Miniport (PPTP)   |
| tp.sys       | c:\windows\system32\drivers\raspp<br>Kernel Driver Yes Manual<br>Running OK Normal<br>No Yes                          |
| processor    | Processor Driver  |
| ssr.sys      | c:\windows\system32\drivers\proce<br>Kernel Driver No Manual<br>Stopped OK Normal<br>No No                            |

|         |   |
|---------|---|
| ptilink | Direct Parallel Link Driver   |
| .sys    | c:\windows\system32\drivers\ptilink<br>Kernel Driver Yes Manual<br>Running OK Normal<br>No Yes Available Kernel |
| ql1080  | ql1080 Not Available Kernel<br>Driver No Disabled Stopped<br>OK Normal No                                       |
| ql10wnt | Ql10wnt Not Available Kernel<br>Driver No Disabled Stopped<br>OK OK Normal No                                   |
| ql12160 | ql12160 Not Available Kernel<br>Driver No Disabled Stopped<br>OK OK Normal No                                   |
| ql1240  | ql1240 Not Available Kernel<br>Driver No Disabled Stopped<br>OK OK Normal No                                    |
| ql1280  | ql1280 Not Available Kernel<br>Driver No Disabled Stopped<br>OK OK Normal No                                    |
| ql2100  | ql2100 Not Available Kernel<br>Driver No Disabled Stopped<br>OK OK Normal No                                    |
| ql2200  | ql2200 Not Available Kernel<br>Driver No Disabled Stopped<br>OK OK Normal No                                    |
| ql2300  | ql2300 Not Available Kernel<br>Driver No Disabled Stopped<br>OK OK Normal No                                    |
| o.sys   | c:\windows\system32\drivers\ql230<br>Kernel Driver Yes Boot<br>Running OK Normal<br>No Yes                      |
| qlvika  | qlvika  |
| .sys    | c:\windows\system32\drivers\qlvika<br>Kernel Driver No Auto<br>Stopped OK Normal<br>No No                       |
| rasacd  | Remote Access Auto Connection<br>Driver   |
| d.sys   | c:\windows\system32\drivers\rasac<br>Kernel Driver Yes System<br>Running OK Normal<br>No Yes                    |
| rasl2tp | WAN Miniport (L2TP)   |
| p.sys   | c:\windows\system32\drivers\rasl2t<br>Kernel Driver Yes Manual<br>Running OK Normal<br>No Yes                   |
| rasppoe | Remote Access PPPOE Driver  |
| poe.sys | c:\windows\system32\drivers\raspp<br>Kernel Driver Yes Manual<br>Running OK Normal<br>No Yes                    |
| raspti  | Direct Parallel   |
| .sys    | c:\windows\system32\drivers\raspti<br>Kernel Driver Yes Manual<br>Running OK Normal<br>No Yes                   |
| rdbss   | Rdbss   |
| sys     | c:\windows\system32\drivers\rdbss.<br>File System Driver Yes<br>System Running OK<br>Normal No Yes              |

```

rdpcdd RDPcDD
c:\windows\system32\drivers\rdpcd
d.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\rdpw
d.sys Kernel Driver Yes Manual
Running OK Ignore
No Yes
secdrv Secdrv
c:\windows\system32\drivers\secdr
v.sys Kernel Driver No Manual
Stopped OK Normal
No No
serenum Serenum Filter Driver
c:\windows\system32\drivers\seren
um.sys Kernel Driver Yes Manual
Running OK Normal
No Yes
serial Serial port driver
c:\windows\system32\drivers\serial.
sys Kernel Driver Yes System
Running OK Ignore
No Yes
sfloppy Sfloppy
c:\windows\system32\drivers\sflopp
y.sys Kernel Driver No System
Stopped OK Ignore
No No
simbad Simbad Not Available Kernel
Driver No Disabled Stopped
OK Normal No
sparrow Sparrow Not Available Kernel
Driver No Disabled Stopped
OK Normal No
srv Srv
c:\windows\system32\drivers\srv.sy
s File System Driver Yes
Manual Running OK
Normal No Yes
swenum Software Bus Driver
c:\windows\system32\drivers\swen
um.sys Kernel Driver Yes Manual
Running OK Normal
No Yes
symc810 symc810 Not Available Kernel
Driver No Disabled Stopped
OK Normal No
symc8xx symc8xx Not Available Kernel
Driver No Disabled Stopped
OK Normal No
symmpi symmpi Not Available Kernel
Driver No Disabled Stopped
OK Normal No
No
sym_hi sym_hi Not Available Kernel
Driver No Disabled Stopped
OK Normal No
No
sym_u3 sym_u3 Not Available Kernel
Driver No Disabled Stopped
OK Normal No
No

```

```

tcPIP TCP/IP Protocol Driver
c:\windows\system32\drivers\tcpip.
sys Kernel Driver Yes System
Running OK Normal
tdpipe TdPIPE Yes
c:\windows\system32\drivers\tdpip
e.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\termd
d.sys Kernel Driver Yes System
Running OK Normal
No Yes
toside TosIde Not Available Kernel
Driver No Disabled Stopped
OK Normal No
No
udfs Udfs
c:\windows\system32\drivers\udfs.s
ys File System Driver No
Disabled Stopped OK
Normal No No
ultra ultra Not Available Kernel
Driver No Disabled Stopped
OK Normal No
No
update Microcode Update Driver
c:\windows\system32\drivers\updat
e.sys Kernel Driver Yes Manual
Running OK Normal
No Yes
usbhci Microsoft USB 2.0 Enhanced Host
Controller Miniport Driver
c:\windows\system32\drivers\usbeh
ci.sys Kernel Driver Yes Manual
Running OK Normal
No Yes
usbhub USB2 Enabled Hub
c:\windows\system32\drivers\usbhu
b.sys Kernel Driver Yes Manual
Running OK Normal
No Yes
usbhcci Microsoft USB Open Host Controller
Miniport Driver
c:\windows\system32\drivers\usboh
ci.sys Kernel Driver No Manual
Stopped OK Normal
No No
usbstor USB Mass Storage Driver
c:\windows\system32\drivers\usbst
or.sys Kernel Driver No Manual
Stopped OK Normal
No No
usbuhci Microsoft USB Universal Host
Controller Miniport Driver
c:\windows\system32\drivers\usbuh
ci.sys Kernel Driver Yes Manual
Running OK Normal
No Yes
vgasave VGA Display Controller.
c:\windows\system32\drivers\vga.s
ys Kernel Driver Yes System
Running OK Ignore
No Yes
viaide ViaIde Not Available Kernel
Driver No Disabled Stopped
OK Normal No
No

```

```

volsnap Storage volumes
c:\windows\system32\drivers\volsn
ap.sys Kernel Driver Yes Boot
Running OK Normal
wanarp Remote Access ARP Driver
c:\windows\system32\drivers\wana
rp.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
x86perfsys Low Overhead Profiler
\??\c:\lop1.1\x86perfsys.sys
Kernel Driver No Manual
Stopped OK Ignore
No No

```

[Signed Drivers]

| Device Name                              | Signed | Device Class | Driver Version | Driver Date | Manufacturer                          | INF Name                  | Driver Name   | Device ID     |
|--|--------|--------------|----------------|-------------|---------------------------------------|---------------------------|---------------|---------------|
| Microsoft System Management BIOS Driver  | Yes    | SYSTEM       | 5.2.3790.1830  | 10/1/2002   | (Standard system machine.inf)         | ROOT\SYSTEM\0002          | Not Available | Not Available |
| Microcode Update Device                  | Yes    | SYSTEM       | 5.2.3790.0     | 10/1/2002   | (Standard system devices) machine.inf | ROOT\SYSTEM\0001          | Not Available | Not Available |
| Plug and Play Software Device Enumerator | Yes    | SYSTEM       | 5.2.3790.0     | 10/1/2002   | (Standard system devices) machine.inf | ROOT\SYSTEM\0000          | Not Available | Not Available |
| Terminal Server Mouse Driver             | Yes    | SYSTEM       | 5.2.3790.0     | 10/1/2002   | (Standard system machine.inf)         | ROOT\RDP_MOU\0000         | Not Available | Not Available |
| Terminal Server Keyboard Driver          | Yes    | SYSTEM       | 5.2.3790.0     | 10/1/2002   | (Standard system machine.inf)         | ROOT\RDP_KBD\0000         | Not Available | Not Available |
| Terminal Server Device Redirector        | Yes    | SYSTEM       | 5.2.3790.0     | 10/1/2002   | (Standard system machine.inf)         | ROOT\RDPDR\0000           | Not Available | Not Available |
| Direct Parallel Available                | Yes    | NET          | 5.2.3790.0     | 10/1/2002   | Microsoft netrasa.inf                 | ROOT\MS_PPTMINIPORT\0000  | Not Available | Not Available |
| WAN Miniport (PPTP) Available            | Yes    | NET          | 5.2.3790.0     | 10/1/2002   | Microsoft netrasa.inf                 | ROOT\MS_PPTPMINIPORT\0000 | Not Available | Not Available |



WAN Miniport (PPPOE) Yes NET  
5.2.3790.0 10/1/2002  
Microsoft netrasa.inf Not  
Available ROOT\MS\_PPP0EMINI\PORT\0000

WAN Miniport (IP) Yes NET  
5.2.3790.0 10/1/2002  
Microsoft netrasa.inf Not  
Available ROOT\MS\_NDISWANIP\0000

WAN Miniport (Network Monitor) Yes  
NET 5.2.3790.1830  
10/1/2002 Microsoft  
netrasa.inf Not Available  
ROOT\MS\_NDISWANBH\0000

WAN Miniport (L2TP) Yes NET  
5.2.3790.0 10/1/2002  
Microsoft netrasa.inf Not  
Available ROOT\MS\_L2TPMINI\PORT\0000

Video Codecs Yes MEDIA  
5.2.3790.0 10/1/2002  
(Standard system devices) wave.inf  
Not Available  
ROOT\MEDIA\MS\_MMVID

Legacy Video Capture Devices Yes  
MEDIA 5.2.3790.0  
10/1/2002 (Standard system  
devices) wave.inf Not Available  
ROOT\MEDIA\MS\_MMVCD

Media Control Devices Yes MEDIA  
5.2.3790.0 10/1/2002  
(Standard system devices) wave.inf  
Not Available  
ROOT\MEDIA\MS\_MMMCI

Legacy Audio Drivers Yes MEDIA  
5.2.3790.0 10/1/2002  
(Standard system devices) wave.inf  
Not Available  
ROOT\MEDIA\MS\_MMDRV

Audio Codecs Yes MEDIA  
5.2.3790.0 10/1/2002  
(Standard system devices) wave.inf  
Not Available  
ROOT\MEDIA\MS\_MMACM

Low Overhead Profiler Not Available  
LEGACYDRIVER Not  
Available Not Available Not Available Not  
Available Not Available  
ROOT\LEGACY\_X86PERFSYS\0000

Remote Access IP ARP Driver Not  
Available LEGACYDRIVER Not  
Available Not Available Not Available Not  
Available Not Available  
ROOT\LEGACY\_WANARP\0000

volsnap Not Available LEGACYDRIVER  
Not Available Not Available Not  
Available Not Available Not Available  
ROOT\LEGACY\_VOLS\NAP\0000

VGA Display Controller. Not Available  
LEGACYDRIVER Not  
Available Not Available Not Available Not  
Available Not Available  
ROOT\LEGACY\_VGASAVE\0000

TDTCP Not Available LEGACYDRIVER  
Not Available Not Available Not  
Available Not Available Not Available  
ROOT\LEGACY\_T\DTCP\0000

TCP/IP Protocol Driver Not Available  
LEGACYDRIVER Not  
Available Not Available Not Available Not  
Available Not Available  
ROOT\LEGACY\_TCPIP\0000

sacdrv Not Available LEGACYDRIVER  
Available Not Available Not Available Not  
ROOT\LEGACY\_SACDRV\0000

RDPWD Not Available LEGACYDRIVER  
Not Available Not Available Not  
Available Not Available Not Available  
ROOT\LEGACY\_RDPWD\0000

RDPCCD Not Available LEGACYDRIVER  
Not Available Not Available Not  
Available Not Available Not Available  
ROOT\LEGACY\_RDPCCD\0000

Remote Access Auto Connection Driver Not  
Available LEGACYDRIVER Not  
Available Not Available Not Available Not  
Available Not Available  
ROOT\LEGACY\_RASACD\0000

ql2300 Not Available LEGACYDRIVER  
Available Not Available Not Available Not  
ROOT\LEGACY\_QL2300\0000

Page Tracker1 for X86perfsys Not  
Available LEGACYDRIVER Not  
Available Not Available Not Available Not  
Available Not Available  
ROOT\LEGACY\_PGTRACKR01\0000

Partition Manager Not Available  
LEGACYDRIVER Not  
Available Not Available Not Available Not  
Available Not Available  
ROOT\LEGACY\_PARTMGR\0000

Null Not Available LEGACYDRIVER  
Not Available Not Available Not  
Available Not Available Not Available  
ROOT\LEGACY\_NULL\0000

Network Monitor Driver Not Available  
LEGACYDRIVER Not  
Available Not Available Not Available Not  
Available Not Available  
ROOT\LEGACY\_NM\0000

NetBios over Tcpi Not Available  
LEGACYDRIVER Not  
Available Not Available Not Available Not  
Available Not Available  
ROOT\LEGACY\_NETBT\0000

NDProxy Not Available LEGACYDRIVER  
Not Available Not Available Not  
Available Not Available Not Available  
ROOT\LEGACY\_NDPROXY\0000

NDIS Usermode I/O Protocol Not  
Available LEGACYDRIVER Not  
Available Not Available Not Available Not  
Available Not Available  
ROOT\LEGACY\_NDISUIO\0000

Remote Access NDIS TAPI Driver Not  
Available LEGACYDRIVER Not  
Available Not Available Not Available Not  
Available Not Available  
ROOT\LEGACY\_NDIS\TAPI\0000

NDIS System Driver Not Available  
LEGACYDRIVER Not  
Available Not Available Not Available Not  
Available Not Available  
mountmgr Not Available LEGACYDRIVER  
Not Available Not Available Not  
Available Not Available Not Available  
ROOT\LEGACY\_MOUNTMGR\0000

modem Not Available LEGACYDRIVER  
Not Available Not Available Not  
Available Not Available Not Available  
ROOT\LEGACY\_MODEM\0000

mnmd Not Available LEGACYDRIVER  
Available Not Available Not Available Not  
ROOT\LEGACY\_MNMDD\0000

ksecdd Not Available LEGACYDRIVER  
Available Not Available Not Available Not  
ROOT\LEGACY\_KSE\CCD\0000

IPSEC driver Not Available LEGACYDRIVER  
Available Not Available Not Available Not  
ROOT\LEGACY\_IPSEC\0000

IntellIde Not Available LEGACYDRIVER  
Available Not Available Not Available Not  
ROOT\LEGACY\_INTELIDE\0000

HTTP Not Available LEGACYDRIVER  
Available Not Available Not Available Not  
ROOT\LEGACY\_HTTP\0000

Generic Packet Classifier Not Available  
LEGACYDRIVER Not  
Available Not Available Not Available Not  
Available Not Available  
ROOT\LEGACY\_GPC\0000

Fips Not Available LEGACYDRIVER  
Available Not Available Not Available Not  
ROOT\LEGACY\_FIPS\0000

dmload Not Available LEGACYDRIVER  
Available Not Available Not Available Not  
ROOT\LEGACY\_DMLOAD\0000

dmboot Not Available LEGACYDRIVER  
Available Not Available Not Available Not  
ROOT\LEGACY\_DMBOOT\0000

CRC Disk Filter Driver Not Available  
LEGACYDRIVER Not  
Available Not Available Not Available Not  
Available Not Available  
ROOT\LEGACY\_CRCDISK\0000

Beep Not Available LEGACYDRIVER  
Available Not Available Not Available Not  
ROOT\LEGACY\_BEEP\0000

Altiris Kernel Driver Not Available  
 LEGACYDRIVER Not Available  
 Available Not Available Not Available Not Available  
 Available Not Available  
 ROOT\LEGACY\_ALKERNEL\0000

AFD Networking Support Environment Not Available  
 LEGACYDRIVER Not Available  
 Available Not Available Not Available Not Available  
 Available Not Available  
 ROOT\LEGACY\_AFD\0000

Generic volume Yes VOLUME  
 5.2.3790.1830  
 10/1/2002 Microsoft  
 volume.inf Not Available  
 STORAGE\VOLUME\1&30A96598&0  
 &SIGNATURE1A321A31OFFSET7E00LENGTH12A1  
 C90400

Volume Manager Yes SYSTEM  
 5.2.3790.0 10/1/2002  
 (Standard system devices)  
 machine.inf Not Available  
 ROOT\FTDISK\0000

Logical Disk Manager Yes SYSTEM  
 5.2.3790.0 10/1/2002  
 (Standard system devices)  
 machine.inf Not Available  
 ROOT\DMIO\0000

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

ACPI Power Button Yes SYSTEM  
 5.2.3790.1830  
 10/1/2002 (Standard system devices)  
 machine.inf Not Available  
 ACPI\PNP0C0C\3&61AAA01&0

Intel(R) 82801EB SMBus Controller - 24D3  
 Yes SYSTEM  
 7.0.0.1011 1/10/2005 Intel  
 ich5core.inf Not Available  
 PCI\VEN\_8086&DEV\_24D3&SUBSYS  
 \_3208103C&REV\_02\3&61AAA01&0&FB

Secondary IDE Channel Yes HDC  
 5.2.3790.1830  
 10/1/2002 (Standard IDE  
 ATA/ATAPI controllers) mshdc.inf Not  
 Available  
 PCIIDE\IDECHANNEL\4&1D65F1F&  
 0&1

Disk drive Yes DISKDRIVE  
 5.2.3790.0 10/1/2002  
 (Standard disk drives) disk.inf  
 Not Available  
 IDE\DISKMAXTOR\_6L080M0\_\_\_\_\_  
 BANC1G10\324C4E3033  
 41474B20202020202020202020202020

Primary IDE Channel Yes HDC  
 5.2.3790.1830  
 10/1/2002 (Standard IDE  
 ATA/ATAPI controllers) mshdc.inf Not  
 Available  
 PCIIDE\IDECHANNEL\4&1D65F1F&  
 0&0

Intel(R) 82801EB Ultra ATA Storage Controllers  
 Yes HDC  
 6.3.0.1005 11/17/2004 Intel  
 ich5side.inf Not Available  
 PCI\VEN\_8086&DEV\_24D1&SUBSYS  
 \_3208103C&REV\_02\3&61AAA01&0&FA

Generic Bus Yes SYSTEM  
 5.2.3790.1830  
 10/1/2002 (Standard system  
 devices) machine.inf Not Available

Communications ports Yes PORTS  
 5.2.3790.0 10/1/2002  
 (Standard port types)  
 msports.inf Not Available  
 ACPI\PNP0501\1

Generic Bus Yes SYSTEM  
 5.2.3790.1830  
 10/1/2002 (Standard system  
 devices) machine.inf Not Available  
 ACPI\PNP0A05\1

PS/2 Compatible Mouse Yes MOUSE  
 5.2.3790.1830  
 10/1/2002 Microsoft  
 msmouse.inf Not Available  
 ACPI\PNP0F13\4&369939D9&0

Standard 101/102-Key or Microsoft Natural PS/2  
 Keyboard Yes KEYBOARD  
 5.2.3790.0 10/1/2002  
 (Standard keyboards)  
 keyboard.inf Not Available  
 ACPI\PNP0303\4&369939D9&0

System timer Yes SYSTEM  
 5.2.3790.1830  
 10/1/2002 (Standard system  
 devices) machine.inf Not Available  
 ACPI\PNP0100\4&369939D9&0

System speaker Yes SYSTEM  
 5.2.3790.1830  
 10/1/2002 (Standard system  
 devices) machine.inf Not Available  
 ACPI\PNP0800\4&369939D9&0

System CMOS/real time clock Yes  
 SYSTEM 5.2.3790.1830  
 10/1/2002 (Standard system  
 devices) machine.inf Not Available  
 ACPI\PNP0B00\4&369939D9&0

Programmable interrupt controller Yes  
 SYSTEM 5.2.3790.1830  
 10/1/2002 (Standard system  
 devices) machine.inf Not Available  
 ACPI\PNP0000\4&369939D9&0

Numeric data processor Yes SYSTEM  
 5.2.3790.1830  
 10/1/2002 (Standard system  
 devices) machine.inf Not Available  
 ACPI\PNP0C04\4&369939D9&0

Direct memory access controller Yes  
 SYSTEM 5.2.3790.1830  
 10/1/2002 (Standard system  
 devices) machine.inf Not Available  
 ACPI\PNP0200\4&369939D9&0

Motherboard resources Yes SYSTEM  
 5.2.3790.1830  
 10/1/2002 (Standard system  
 devices) machine.inf Not Available  
 ACPI\PNP0C02\1F

ISAPNP Read Data Port Yes SYSTEM  
 5.2.3790.0 10/1/2002  
 (Standard system devices)  
 machine.inf Not Available  
 ISAPNP\READDATAPORT\0

Intel(R) 82801EB LPC Interface Controller - 24D0  
 Yes SYSTEM  
 7.0.0.1011 1/10/2005 Intel  
 ich5core.inf Not Available  
 PCI\VEN\_8086&DEV\_24D0&SUBSYS  
 \_00000000&REV\_02\3&61AAA01&0&F8

Monitor Yes MONITOR  
 5.1.2001.0  
 6/6/2001 (Standard monitor  
 types) monitor.inf Not Available  
 DISPLAY\DEFAULT\_MONITOR\5&B  
 6946BC&0&80000000&07&01

RAGE XL PCI Family (MicroSoft Corporation)  
 Yes DISPLAY  
 5.10.2600.6014  
 8/8/2001 ATI Technologies Inc.  
 atiixpad.inf Not Available  
 PCI\VEN\_1002&DEV\_4752&SUBSYS  
 \_3208103C&REV\_27\4&3A321F3&0&08F0

Intel(R) 82801 PCI Bridge - 244E Yes  
 SYSTEM 7.0.0.1011  
 1/10/2005 Intel  
 dmi\_pci.inf Not Available  
 PCI\VEN\_8086&DEV\_244E&SUBSYS  
 \_00000000&REV\_C2\3&61AAA01&0&F0

USB Root Hub Yes USB  
 5.2.3790.1830  
 10/1/2002 (Standard USB Host  
 Controller) usbport.inf Not Available  
 USB\ROOT\_HUB2\4&D12AFCF&0

Intel(R) 82801EB USB Enhanced Host Controller  
 - 24DD Yes USB  
 6.3.0.1005 11/17/2004 Intel  
 ich5usb.inf Not Available  
 PCI\VEN\_8086&DEV\_24DD&SUBSY  
 S\_3208103C&REV\_02\3&61AAA01&0&EF

USB Root Hub Yes USB  
 5.2.3790.1830  
 10/1/2002 (Standard USB Host  
 Controller) usbport.inf Not Available  
 USB\ROOT\_HUB\4&2F4E9900&0

Intel(R) 82801EB USB Universal Host Controller -  
 24D4 Yes USB  
 6.3.0.1005 11/17/2004 Intel  
 ich5usb.inf Not Available  
 PCI\VEN\_8086&DEV\_24D4&SUBSYS  
 \_3208103C&REV\_02\3&61AAA01&0&E9

USB Root Hub Yes USB  
 5.2.3790.1830  
 10/1/2002 (Standard USB Host  
 Controller) usbport.inf Not Available  
 USB\ROOT\_HUB\4&27E207BF&0

Intel(R) 82801EB USB Universal Host Controller -  
 24D2 Yes USB  
 6.3.0.1005 11/17/2004 Intel  
 ich5usb.inf Not Available  
 PCI\VEN\_8086&DEV\_24D2&SUBSYS  
 \_3208103C&REV\_02\3&61AAA01&0&E8

Intel(R) 6700PXH I/OxAPIC Interrupt Controller B  
 - 0327 Yes SYSTEM  
 6.3.0.1005 11/17/2004 Intel  
 e7520.inf Not Available  
 PCI\VEN\_8086&DEV\_0327&SUBSYS  
 \_3208103C&REV\_09\4&39D7C96F&0&0330

```

Intel(R) 6700PXH PCI Express-to-PCI Bridge B -
032A Yes SYSTEM
6.3.0.1005 11/17/2004 Intel
e7520.inf Not Available
PCI\VEN_8086&DEV_032A&SUBSYS
_00000000&REV_09\4&39D7C96F&0&0230

Intel(R) 6700/6702PXH I/OxAPIC Interrupt
Controller A - 0326 Yes SYSTEM
6.3.0.1005 11/17/2004 Intel
e7520.inf Not Available
PCI\VEN_8086&DEV_0326&SUBSYS
_3208103C&REV_09\4&39D7C96F&0&0130

NDIS Client For Broadcom NetXtreme II GigE
Available Not Available NET Not
Available Not Available Not Available Not
Available Not Available
B06BDRV\L2ND&PCI_164A14E4&S
UBSYS_3101103C&REV_02\6&D50AD48&0&2005
0501
Broadcom BCM5706C NetXtreme II GigE No
SYSTEM 2.4.999.0
11/3/2005 Broadcom Corporation
oem5.inf Not Available
PCI\VEN_14E4&DEV_164A&SUBSYS
_3101103C&REV_02\5&55F0281&0&080030

Intel(R) 6700PXH PCI Express-to-PCI Bridge A -
0329 Yes SYSTEM
6.3.0.1005 11/17/2004 Intel
e7520.inf Not Available
PCI\VEN_8086&DEV_0329&SUBSYS
_00000000&REV_09\4&39D7C96F&0&0030

Intel(R) E7520 PCI Express Root Port C0 - 3599
Yes SYSTEM
6.3.0.1005 11/17/2004 Intel
e7520.inf Not Available
PCI\VEN_8086&DEV_3599&SUBSYS
_00000000&REV_0C\3&61AAA01&0&830
Broadcom NetXtreme Gigabit Ethernet Yes
NET 7.86.0.0
8/23/2004 Broadcom oem3.inf
Not Available
PCI\VEN_14E4&DEV_1659&SUBSYS
_1659103C&REV_11\4&1C834E48&0&0028

Intel(R) E7520 PCI Express Root Port B1 - 3598
Yes SYSTEM
6.3.0.1005 11/17/2004 Intel
e7520.inf Not Available
PCI\VEN_8086&DEV_3598&SUBSYS
_00000000&REV_0C\3&61AAA01&0&828
Broadcom NetXtreme Gigabit Ethernet Yes
NET 7.86.0.0
8/23/2004 Broadcom oem3.inf
Not Available
PCI\VEN_14E4&DEV_1659&SUBSYS
_1659103C&REV_11\4&253DB27D&0&0020

Intel(R) E7525/E7520 PCI Express Root Port B0 -
3597 Yes SYSTEM
6.3.0.1005 11/17/2004 Intel
e7520.inf Not Available
PCI\VEN_8086&DEV_3597&SUBSYS
_00000000&REV_0C\3&61AAA01&0&820
Intel(R) E7525/E7520/E7320 PCI Express Root
Port A0 - 3595 Yes SYSTEM
6.3.0.1005 11/17/2004 Intel
e7520.inf Not Available
PCI\VEN_8086&DEV_3595&SUBSYS
_00000000&REV_0C\3&61AAA01&0&810

```

```

Intel(R) E7525/E7520 Error Reporting Registers -
3591 Yes SYSTEM
6.3.0.1005 11/17/2004 Intel
e7520.inf Not Available
PCI\VEN_8086&DEV_3591&SUBSYS
_00000000&REV_0C\3&61AAA01&0&810

Intel(R) Processor 3590 Yes SYSTEM
6.3.0.1005 11/17/2004 Intel
e7520.inf Not Available
PCI\VEN_8086&DEV_3590&SUBSYS
_00000000&REV_0C\3&61AAA01&0&800
PCI bus Yes SYSTEM
5.2.3790.1830
10/1/2002 (Standard system
devices) machine.inf Not Available
ACPI\PNP0A03\2&DABA3FF&0

Intel Processor Yes
PROCESSOR 5.2.3790.1830
10/1/2002 Intel cpu.inf
Not Available
ACPI\GENUINEINTEL_-
_X86_FAMILY_15_MODEL_4\3
Intel Processor Yes
PROCESSOR 5.2.3790.1830
10/1/2002 Intel cpu.inf
Not Available
ACPI\GENUINEINTEL_-
_X86_FAMILY_15_MODEL_4\2
Intel Processor Yes
PROCESSOR 5.2.3790.1830
10/1/2002 Intel cpu.inf
Not Available
ACPI\GENUINEINTEL_-
_X86_FAMILY_15_MODEL_4\1
Intel Processor Yes
PROCESSOR 5.2.3790.1830
10/1/2002 Intel cpu.inf
Not Available
ACPI\GENUINEINTEL_-
_X86_FAMILY_15_MODEL_4\0
Microsoft ACPI-Compliant System Yes
SYSTEM 5.2.3790.0
10/1/2002 Microsoft acpi.inf
Not Available ACPI_HAL\PNP0C08\0

ACPI Multiprocessor PC Yes
COMPUTER 5.2.3790.0
10/1/2002 (Standard computers)
hal.inf Not Available
ROOT\ACPI_HAL\0000
Not Available Not Available Not Available Not
Available Not Available Not Available Not
Available Not Available HTREE\ROOT\0

[Environment Variables]

Variable Value User Name
ComSpec %SystemRoot%\system32\cmd.exe
<SYSTEM>
Path
%SystemRoot%\system32;%Syste
mRoot%;%SystemRoot%\System32\Wbem;C:\Pr
ogram 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 4
Stepping 3, GenuineIntel <SYSTEM>

```

```

PROCESSOR_REVISION 0403
NUMBER_OF_PROCESSORS 4
<SYSTEM>
ClusterLog C:\WINDOWS\Cluster\cluster.log
<SYSTEM>
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.J
S;.JSE;.WSF;.WSH <SYSTEM>
TEMP %SystemRoot%\TEMP
<SYSTEM>
TMP %SystemRoot%\TEMP
<SYSTEM>
FP_NO_HOST_CHECK NO
<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
DL18\Administrator

TMP %USERPROFILE%\Local
Settings\Temp
DL18\Administrator

[Print Jobs]

Document Size Owner Notify
Status Time Submitted
Start Time Until Time Elapsed
Time Pages Printed Job ID
Priority Parameters Driver
Print Processor Host
Print Queue Data Type Name

[Network Connections]

Local Name Remote Name Type
Status User Name

[Running Tasks]

Name Path Process ID Priority
Min Working Set Max
Working Set Start Time Version Size
File Date
system idle process Not Available 0
0 Not Available Not
Available Not Available Not
Available Not Available
system Not Available 4 8
0 1413120 Not
Available Not Available Not
Available Not
smss.exe Not Available 524 11
204800 1413120
7/7/2006 3:26 PM Not
Available Not Available Not Available

```

```

csrss.exe Not Available 588 13
Not Available Not Available
7/7/2006 3:26 PM Not
Available Not Available Not Available
winlogon.exe c:\windows\system32\winlogon.exe
628 13 204800
1413120 7/7/2006 3:26 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 497.00
KB (508,928 bytes) 4/14/2005 10:00 AM

services.exe c:\windows\system32\services.exe
672 9 204800
1413120 7/7/2006 3:26 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 107.50
KB (110,080 bytes) 4/12/2005 1:43 PM

lsass.exe c:\windows\system32\lsass.exe
684 9 204800
1413120 7/7/2006 3:26 PM
5.2.3790.0 (srv03_rtm.030324-
2048) 13.00 KB (13,312 bytes)
4/12/2005 1:42 PM
svchost.exe c:\windows\system32\svchost.exe
872 8 204800
1413120 7/7/2006 3:26 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:01 AM

svchost.exe Not Available 968 8
Not Available Not Available
7/7/2006 3:26 PM Not
Available Not Available Not Available
svchost.exe Not Available 1060 8
Not Available Not Available
7/7/2006 3:26 PM Not
svchost.exe Not Available 1136 8
Not Available Not Available
7/7/2006 3:26 PM Not
Available Not Available Not Available
svchost.exe c:\windows\system32\svchost.exe
1148 8 204800
1413120 7/7/2006 3:26 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:01 AM

spoolsv.exe c:\windows\system32\spoolsv.exe
1988 8 204800
1413120 7/7/2006 3:26 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 57.00 KB
(58,368 bytes) 4/14/2005 10:01 AM

msdtc.exe Not Available 2020 8
Not Available Not Available
7/7/2006 3:26 PM Not
Available Not Available Not Available
aclient.exe c:\program
files\altiris\aclient\aclient.exe 356
8 204800 1413120
7/7/2006 3:26 PM 6.5.241
4.75 MB (4,984,908 bytes)
5/3/2005 8:59 AM
svchost.exe c:\windows\system32\svchost.exe
448 8 204800
1413120 7/7/2006 3:26 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:01 AM

```

```

inetinfo.exe c:\windows\system32\inetinfo
fo.exe 508 8 204800
1413120 7/7/2006 3:26 PM
6.0.3790.1830
(srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:02 AM
svchost.exe Not Available 748 8
Not Available Not Available
7/7/2006 3:26 PM Not
Available Not Available Not Available
rsys.exe Not Available 1048 8
Not Available Not Available
7/7/2006 3:26 PM Not
Available Not Available Not Available
svchost.exe c:\windows\system32\svchost.exe
1780 8 204800
1413120 7/7/2006 3:26 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:01 AM

svchost.exe c:\windows\system32\svchost.exe
1960 8 204800
1413120 7/7/2006 3:26 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:01 AM

wmiprvse.exe Not Available 1404
8 Not Available Not
Available 7/7/2006 3:27 PM Not
Available Not Available Not Available
csrss.exe Not Available 1464 13
Not Available Not Available
7/7/2006 3:33 PM Not
Available Not Available Not Available
winlogon.exe c:\windows\system32\winlogon.exe
1480 13 204800
1413120 7/7/2006 3:33 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 497.00
KB (508,928 bytes) 4/14/2005 10:00 AM

rdpclip.exe c:\windows\system32\rdpclip.exe
1716 8 204800
1413120 7/7/2006 3:33 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 68.00 KB
(69,632 bytes) 4/14/2005 10:01 AM

explorer.exe c:\windows\explorer.exe 2164
8 204800 1413120
7/7/2006 3:33 PM
6.00.3790.1830
(srv03_sp1_rtm.050324-1447) 1.00 MB
(1,050,624 bytes) 4/14/2005 10:01 AM

aclntusr.exe c:\program
files\altiris\aclient\aclntusr.exe 2416
8 204800 1413120
7/7/2006 3:33 PM 6, 5, 241
180.00 KB (184,320 bytes)
5/3/2005 8:59 AM
svchost.exe c:\windows\system32\svchost.exe
2528 8 204800
1413120 7/7/2006 3:33 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:01 AM

logon.scr Not Available 3468 4
Not Available Not Available
7/7/2006 3:43 PM Not
Available Not Available Not Available

```

```

w3wp.exe c:\windows\system32\inetrv\w3wp
.exe 3376 8 204800
1413120 7/7/2006 3:53 PM
6.0.3790.1830
(srv03_sp1_rtm.050324-1447) 7.00 KB
(7,168 bytes) 4/14/2005 10:01 AM
svchost.exe c:\windows\system32\svchost.exe
3532 8 204800
1413120 7/7/2006 3:53 PM
5.2.3790.0 (srv03_rtm.030324-
2048) 5.50 KB (5,632 bytes)
4/12/2005 1:42 PM
mmc.exe c:\windows\system32\mmc.exe
3836 8 204800
1413120 7/7/2006 4:03 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 829.00
KB (848,896 bytes) 4/14/2005 10:01 AM

helpctr.exe c:\windows\pchealth\helpctr\binarie
s\helpctr.exe 5980 8 204800
1413120 7/8/2006 11:42 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 778.00
KB (796,672 bytes) 4/14/2005 10:01 AM

helpsvc.exe c:\windows\pchealth\helpctr\binarie
s\helpsvc.exe 3724 8 204800
1413120 7/8/2006 11:42 PM
5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 745.00
KB (762,880 bytes) 4/14/2005 10:01 AM

wmiprvse.exe Not Available 4900
8 Not Available Not
Available 7/8/2006 11:42 PM Not
Available Not Available Not Available

[Loaded Modules]

Name Version Size File Date
Manufacturer Path
winlogon 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 497.00
KB (508,928 bytes) 4/14/2005 10:00 AM
Microsoft Corporation
c:\windows\system32\winlogon.exe
ntdll 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 748.50
KB (766,464 bytes) 4/12/2005 1:42 PM
Microsoft Corporation
c:\windows\system32\ntdll.dll
kernel32 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 1,014.00
KB (1,038,336 bytes) 4/14/2005 10:01 AM
Microsoft Corporation
c:\windows\system32\kernel32.dll
advapi32 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 605.50
KB (620,032 bytes) 4/12/2005 1:42 PM
Microsoft Corporation
c:\windows\system32\advapi32.dll

```

rpqrt4 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 627.00  
KB (642,048 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\rpqrt4.dll

crypt32 5.131.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 582.00  
KB (595,968 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\crypt32.dll

msasn1 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 56.50 KB  
(57,856 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\msasn1.dll

msvcrt 7.0.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 340.50  
KB (348,672 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\msvcrt.dll

user32 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 574.50  
KB (588,288 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\user32.dll

gdi32 5.2.3790.2606  
(srv03\_sp1\_gdr.051230-1233) 275.00  
KB (281,600 bytes) 12/30/2005 8:12 PM  
Microsoft Corporation  
c:\windows\system32\gdi32.dll

nddeapi 5.2.3790.0 (srv03\_rtm.030324-  
2048) 16.00 KB (16,384 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\nddeapi.dll

profmap 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 22.50 KB  
(23,040 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\profmap.dll

netapi32 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 341.50  
KB (349,696 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\netapi32.dll

userenv 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 771.00  
KB (789,504 bytes) 4/12/2005 1:43 PM  
Microsoft Corporation  
c:\windows\system32\userenv.dll

psapi 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 20.00 KB  
(20,480 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\psapi.dll

regapi 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 55.00 KB  
(56,320 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\regapi.dll

secur32 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 64.00 KB  
(65,536 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\secur32.dll

setupapi 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 1.03 MB  
(1,079,808 bytes) 4/12/2005 1:43 PM  
Microsoft Corporation  
c:\windows\system32\setupapi.dll

version 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 18.00 KB  
(18,432 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\version.dll

winsta 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 54.50 KB  
(55,808 bytes) 4/14/2005 10:00 AM  
Microsoft Corporation  
c:\windows\system32\winsta.dll

ws2\_32 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 82.00 KB  
(83,968 bytes) 4/14/2005 10:00 AM  
Microsoft Corporation  
c:\windows\system32\ws2\_32.dll

ws2help 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 19.50 KB  
(19,968 bytes) 4/14/2005 10:00 AM  
Microsoft Corporation  
c:\windows\system32\ws2help.dll

msgina 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 1.16 MB  
(1,211,904 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\msgina.dll

shsvcs 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 131.50  
KB (134,656 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\shsvcs.dll

shlwapi 6.00.3790.2564  
(srv03\_sp1\_gdr.051104-1524) 314.00  
KB (321,536 bytes) 11/7/2005 2:19 AM  
Microsoft Corporation  
c:\windows\system32\shlwapi.dll

sfc 5.2.3790.0 (srv03\_rtm.030324-  
2048) 4.50 KB (4,608 bytes)  
4/12/2005 1:43 PM  
Microsoft Corporation  
c:\windows\system32\sfc.dll

sfc\_os 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 138.00  
KB (141,312 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\sfc\_os.dll

wintrust 5.131.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 162.00  
KB (165,888 bytes) 4/14/2005 10:00 AM  
Microsoft Corporation  
c:\windows\system32\wintrust.dll

imagehlp 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 145.50  
KB (148,992 bytes) 4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\imagehlp.dll

ole32 5.2.3790.2492  
(srv03\_sp1\_gdr.050720-1521) 1.19 MB  
(1,245,184 bytes) 7/20/2005 8:24 PM  
Microsoft Corporation  
c:\windows\system32\ole32.dll

comctl32 6.0 (srv03\_sp1\_rtm.050324-1447)  
1.00 MB (1,051,136 bytes)  
3/24/2005 8:41 PM  
Microsoft Corporation  
c:\windows\winsxs\x86\_microsoft.w  
indows.common-  
controls\_6595b64144ccf1df\_6.0.3790.1830\_x-  
ww\_7ae38ccf\comctl32.dll

winscard 5.2.3790.0 (srv03\_rtm.030324-  
2048) 98.50 KB (100,864 bytes)  
4/12/2005 1:43 PM  
Microsoft Corporation  
c:\windows\system32\wincard.dll

wtsapi32 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 19.00 KB  
(19,456 bytes) 4/14/2005 10:00 AM  
Microsoft Corporation  
c:\windows\system32\wtsapi32.dll

sxs 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 743.50  
KB (761,344 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\sxs.dll

winmm 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 172.50  
KB (176,640 bytes) 4/14/2005 10:00 AM  
Microsoft Corporation  
c:\windows\system32\winmm.dll

shell32 6.00.3790.2534  
(srv03\_sp1\_gdr.050922-2352) 7.99 MB  
(8,379,392 bytes) 9/23/2005 9:50 AM  
Microsoft Corporation  
c:\windows\system32\shell32.dll

wldap32 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 174.50  
KB (178,688 bytes) 4/14/2005 10:00 AM  
Microsoft Corporation  
c:\windows\system32\wldap32.dll

rsaenh 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 183.98  
KB (188,392 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\rsaenh.dll

cscdll 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 100.00  
KB (102,400 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\cscdll.dll

dimntfy 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 19.00 KB  
(19,456 bytes) 4/14/2005 10:03 AM  
Microsoft Corporation  
c:\windows\system32\dimntfy.dll

wlnotify 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 94.50 KB  
(96,768 bytes) 4/14/2005 10:00 AM  
Microsoft Corporation  
c:\windows\system32\wlnotify.dll

mpr 5.2.3790.0 (srv03\_rtm.030324-  
2048) 56.00 KB (57,344 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\mpr.dll

oleaut32 5.2.3790.1830 543.00  
KB (556,032 bytes) 4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\oleaut32.dll

winspool 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 147.00  
KB (150,528 bytes) 4/12/2005 1:43 PM  
Microsoft Corporation  
c:\windows\system32\winspool.drv

comctl32 5.82 (srv03\_sp1\_rtm.050324-1447)  
585.00 KB (599,040 bytes)  
3/24/2005 8:41 PM  
Microsoft Corporation  
c:\windows\winsxs\x86\_microsoft.w  
indows.common-  
controls\_6595b64144ccf1df\_5.82.3790.1830\_x-  
ww\_1b6f474a\comctl32.dll

uxtheme 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 202.00  
KB (206,848 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\uxtheme.dll

mprapi 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 89.00 KB  
(91,136 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\mprapi.dll

activeds 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 194.00  
KB (198,656 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\activeds.dll

adslidpc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 146.00  
KB (149,504 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\adslidpc.dll

credui 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 162.00  
KB (165,888 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\credui.dll

atl 3.05.2283 83.00 KB (84,992  
bytes) 4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\atl.dll

rtutils 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 34.50 KB  
(35,328 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\rtutils.dll

samlib 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 46.50 KB  
(47,616 bytes) 4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\samlib.dll

scredir 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 28.00 KB  
(28,672 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\scredir.dll

clbcatq 2001.12.4720.2492  
(srv03\_sp1\_gdr.050720-1521) 500.00  
KB (512,000 bytes) 7/20/2005 8:24 PM  
Microsoft Corporation  
c:\windows\system32\clbcatq.dll

comres 2001.12.4720.0  
(srv03\_rtm.030324-2048) 778.00 KB (796,672  
bytes) 4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\comres.dll

csoui 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 319.50  
KB (327,168 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\csoui.dll

ntmarta 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 120.50  
KB (123,392 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\ntmarta.dll

rdpsnd 5.2.3790.0 (srv03\_rtm.030324-  
2048) 18.00 KB (18,432 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\rdpsnd.dll

xpsp2res 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 2.76 MB  
(2,897,920 bytes) 4/14/2005 10:03 AM  
Microsoft Corporation  
c:\windows\system32\xpsp2res.dll

msacm32 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 22.00 KB  
(22,528 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\msacm32.drv

msacm32 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 69.50 KB  
(71,168 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\msacm32.dll

imaadp32 5.2.3790.0 (srv03\_rtm.030324-  
2048) 15.50 KB (15,872 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\imaadp32.ac  
m

msadp32 5.2.3790.0 (srv03\_rtm.030324-  
2048) 14.50 KB (14,848 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\msadp32.ac  
m

printui 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 563.00  
KB (576,512 bytes) 4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\printui.dll

cfgmgr32 5.2.3790.0 (srv03\_rtm.030324-  
2048) 17.50 KB (17,920 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\cfgmgr32.dll

msg711 5.2.3790.0 (srv03\_rtm.030324-  
2048) 10.00 KB (10,240 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\msg711.acm

msgsm32 5.2.3790.0 (srv03\_rtm.030324-  
2048) 20.50 KB (20,992 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\msgsm32.ac  
m

cabinet 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 81.50 KB  
(83,456 bytes) 3/24/2005 7:35 PM  
Microsoft Corporation  
c:\windows\system32\cabinet.dll

tssoft32 1.01 9.50 KB (9,728 bytes)  
4/12/2005 1:43 PM DSP  
GROUP, INC. c:\windows\system32\tssoft32.acm

tsd32 1.03 16.50 KB (16,896  
bytes) 4/12/2005 1:43 PM DSP  
GROUP, INC. c:\windows\system32\tsd32.dll

msg723 5.2.3790.1830 120.00  
KB (122,880 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\msg723.acm

msaud32 8.00.00.4487288.00 KB (294,912  
bytes) 4/12/2005 1:43 PM  
Microsoft Corporation  
c:\windows\system32\msaud32.ac  
m

sl\_anet 3.02 84.00 KB (86,016  
bytes) 4/12/2005 1:43 PM Sipro  
Lab Telecom Inc.  
c:\windows\system32\sl\_anet.acm

I3codeca 1, 9, 0, 0305 284.00 KB (290,816  
bytes) 4/12/2005 1:43 PM  
Fraunhofer Institut Integrierte  
Schaltungen IIS  
c:\windows\system32\I3codeca.acm

wbemprox 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 20.50 KB  
(20,992 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\wbem\wbem  
prox.dll

wbemcomn 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 221.00  
KB (226,304 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\wbem\wbem  
comn.dll

wbemsvc 5.2.3790.0 (srv03\_rtm.030324-2048) 42.50 KB (43,520 bytes) 4/13/2005 3:12 PM Microsoft Corporation c:\windows\system32\wbem\wbem

svc.dll

fastprox 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 471.00 KB (482,304 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\fastpr

ox.dll

msvcp60 6.05.2144.0 388.00 KB (397,312 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\msvcp60.dll

ntdsapi 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 71.00 KB (72,704 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\ntdsapi.dll

dnsapi 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 153.50 KB (157,184 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\dnsapi.dll

services 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 107.50 KB (110,080 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\services.exe

ncobjapi 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 36.00 KB (36,864 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\ncobjapi.dll

scesrv 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 327.00 KB (334,848 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\scesrv.dll

authz 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 66.50 KB (68,096 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\authz.dll

umpnpgmr 5.2.3790.2477 (srv03\_sp1\_gdr.050629-1534) 135.50 KB (138,752 bytes) 6/29/2005 7:52 PM Microsoft Corporation c:\windows\system32\umpnpgmr.dll

eventlog 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 67.50 KB (69,120 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\eventlog.dll

cryptnet 5.131.3790.1830 (srv03\_sp1\_rtm.050324-1447) 61.00 KB (62,464 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\cryptnet.dll

sensapi 5.2.3790.0 (srv03\_rtm.030324-2048) 6.00 KB (6,144 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\sensapi.dll

lsass 5.2.3790.0 (srv03\_rtm.030324-2048) 13.00 KB (13,312 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\lsass.exe

lsasrv 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 803.00 KB (822,272 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\lsasrv.dll

samsrv 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 450.50 KB (461,312 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\samsrv.dll

cryptdll 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 32.00 KB (32,768 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\cryptdll.dll

msprivs 5.2.3790.0 (srv03\_rtm.030324-2048) 46.50 KB (47,616 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\msprivs.dll

kerberos 5.2.3790.2464 (srv03\_sp1\_gdr.050613-1636) 341.50 KB (349,696 bytes) 6/14/2005 12:10 PM Microsoft Corporation c:\windows\system32\kerberos.dll

msv1\_0 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 141.00 KB (144,384 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\msv1\_0.dll

iphlpapi 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 92.50 KB (94,720 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\iphlpapi.dll

netlogon 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 409.50 KB (419,328 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\netlogon.dll

w32time 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 222.00 KB (227,328 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\w32time.dll

schannel 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 141.00 KB (144,384 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\schannel.dll

wdigest 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 74.00 KB (75,776 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wdigest.dll

rassfm 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 23.00 KB (23,552 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\rassfm.dll

kdcsvc 5.2.3790.2464 (srv03\_sp1\_gdr.050613-1636) 214.50 KB (219,648 bytes) 6/14/2005 12:10 PM Microsoft Corporation c:\windows\system32\kdcsvc.dll

ntdsa 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 1.45 MB (1,516,032 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\ntdsa.dll

esent 5.2.3790.2555 (srv03\_sp1\_gdr.051024-1524) 1,022.00 KB (1,046,528 bytes) 2/1/2006 11:06 AM Microsoft Corporation c:\windows\system32\esent.dll

ntdsatq 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 29.50 KB (30,208 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\ntdsatq.dll

mswsock 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 250.50 KB (256,512 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\mswsock.dll

scedi 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 186.50 KB (190,976 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\scedi.dll

ws03res 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 793.50 KB (812,544 bytes) 4/14/2005 10:03 AM Microsoft Corporation c:\windows\system32\ws03res.dll

hnetcfg 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 343.50 KB (351,744 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\hnetcfg.dll

wshtcpip 5.2.3790.0 (srv03\_rtm.030324-2048) 18.00 KB (18,432 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\wshtcpip.dll

ipsecsvc 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 180.50 KB (184,832 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\ipsecsvc.dll

oakley 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 264.00  
KB (270,336 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\oakley.dll

winipsec 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 35.50 KB  
(36,352 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\winipsec.dll

pstorsvc 5.2.3790.0 (srv03\_rtm.030324-  
2048) 24.00 KB (24,576 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\pstorsvc.dll

psbase 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 84.00 KB  
(86,016 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\psbase.dll

dssenh 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 139.98  
KB (143,336 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\dssenh.dll

wlbcctrl 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 82.00 KB  
(83,968 bytes) 4/14/2005 10:00 AM  
Microsoft Corporation  
c:\windows\system32\wlbcctrl.dll

w3ssl 6.0.3790.0 (srv03\_rtm.030324-  
2048) 15.00 KB (15,360 bytes)  
4/12/2005 1:43 PM  
Microsoft Corporation  
c:\windows\system32\w3ssl.dll

strmfilt 6.0.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 84.00 KB  
(86,016 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\strmfilt.dll

httpapi 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 24.00 KB  
(24,576 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\httpapi.dll

svchost 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 14.00 KB  
(14,336 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\svchost.exe

rpcss 5.2.3790.2492  
(srv03\_sp1\_gdr.050720-1521) 408.00  
KB (417,792 bytes) 7/20/2005 8:24 PM  
Microsoft Corporation  
c:\windows\system32\rpcss.dll

wzcsvc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 364.50  
KB (373,248 bytes) 4/14/2005 10:00 AM  
Microsoft Corporation  
c:\windows\system32\wzcsvc.dll

wmi 5.2.3790.0 (srv03\_rtm.030324-  
2048) 6.50 KB (6,656 bytes)  
4/12/2005 1:43 PM  
Microsoft Corporation  
c:\windows\system32\wmi.dll

dhcpcsvc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 113.50  
KB (116,224 bytes) 4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\dhcpcsvc.dll

rastls 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 180.00  
KB (184,320 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\rastls.dll

cryptui 5.131.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 496.50  
KB (508,416 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\cryptui.dll

rasapi32 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 239.50  
KB (245,248 bytes) 4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\rasapi32.dll

rasman 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 61.50 KB  
(62,976 bytes) 4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\rasman.dll

tapi32 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 179.50  
KB (183,808 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\tapi32.dll

raschap 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 119.50  
KB (122,368 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\raschap.dll

schedsvc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 197.50  
KB (202,240 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\schedsvc.dll

msidle 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 6.50 KB  
(6,656 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\msidle.dll

audiosrv 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 40.50 KB  
(41,472 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\audiosrv.dll

wiarpc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 32.50 KB  
(33,280 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\wiarpc.dll

wkssvc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 130.00  
KB (133,120 bytes) 4/12/2005 1:43 PM  
Microsoft Corporation  
c:\windows\system32\wkssvc.dll

aelupsvc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 26.00 KB  
(26,624 bytes) 4/14/2005 10:03 AM  
Microsoft Corporation  
c:\windows\system32\aelupsvc.dll

apphelp 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 146.50  
KB (150,016 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\apphelp.dll

cryptsvc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 55.50 KB  
(56,832 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\cryptsvc.dll

certcli 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 227.00  
KB (232,448 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\certcli.dll

vssapi 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 548.00  
KB (561,152 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\vssapi.dll

dmserver 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 25.50 KB  
(26,112 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\dmserver.dll

es 2001.12.4720.2492  
(srv03\_sp1\_gdr.050720-1521) 233.00  
KB (238,592 bytes) 7/20/2005 8:24 PM  
Microsoft Corporation  
c:\windows\system32\es.dll

pchsvc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 39.00 KB  
(39,936 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\pchealth\helpctr\binarie  
s\pchsvc.dll

srvsvc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 93.50 KB  
(95,744 bytes) 4/12/2005 1:43 PM  
Microsoft Corporation  
c:\windows\system32\srvsvc.dll

seclogon 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 18.50 KB  
(18,944 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\seclogon.dll

sens 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 36.50 KB  
(37,376 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\sens.dll



trkwks 5.2.3790.0 (srv03\_rtm.030324-2048) 85.00 KB (87,040 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\trkwks.dll

wmisvc 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 140.00 KB (143,360 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\wmisv

c.dll

wuauerv 5.7.3790.1830 (srv03\_sp1\_rtm.050324-1447) 8.00 KB (8,192 bytes) 4/14/2005 10:03 AM Microsoft Corporation c:\windows\system32\wuauerv.dll

wuaueng 5.8.0.2469 built by: lab01\_n(wmbla) 1.28 MB (1,343,768 bytes) 4/14/2005 10:03 AM Microsoft Corporation c:\windows\system32\wuaueng.dll

advpack 6.00.3790.1830 (srv03\_sp1\_rtm.050324-1447) 98.00 KB (100,352 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\advpack.dll

shfolder 6.00.3790.1830 (srv03\_sp1\_rtm.050324-1447) 24.50 KB (25,088 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\shfolder.dll

winhttp 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 353.00 KB (361,472 bytes) 3/24/2005 8:41 PM Microsoft Corporation c:\windows\winsxs\x86\_microsoft.w

indows.winhttp\_6595b64144ccf1df\_5.1.3790.1830\_x-ww\_74150efb\winhttp.dll

mspatcha 5.2.3790.0 (srv03\_rtm.030324-2048) 29.00 KB (29,696 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\mspatcha.dll

comsvcs 2001.12.4720.2517 (srv03\_sp1\_gdr.050825-1634) 1.19 MB (1,247,744 bytes) 8/26/2005 3:18 PM Microsoft Corporation c:\windows\system32\comsvcs.dll

browser 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 76.50 KB (78,336 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\browser.dll

netrap 5.2.3790.0 (srv03\_rtm.030324-2048) 11.50 KB (11,776 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\netrap.dll

wbemcore 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 497.50 KB (509,440 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\wbem

core.dll

esscli 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 250.00 KB (256,000 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\esscli.

miutils 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 93.50 KB (95,744 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\miuti

ls.dll

repdrvfs 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 172.50 KB (176,640 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\repdrv

fs.dll

wmiprvsd 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 404.00 KB (413,696 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\wmipr

vsd.dll

wbemess 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 271.50 KB (278,016 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\wbem

ess.dll

ncprov 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 46.50 KB (47,616 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\ncprov

.dll

netman 5.2.3790.2516 (srv03\_sp1\_gdr.050824-1616) 258.50 KB (264,704 bytes) 8/25/2005 5:12 PM Microsoft Corporation c:\windows\system32\netman.dll

netshell 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 1.73 MB (1,812,992 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\netshell.dll

clusapi 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 60.00 KB (61,440 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\clusapi.dll

wininet 6.00.3790.2564 (srv03\_sp1\_gdr.051104-1524) 647.00 KB (662,528 bytes) 11/7/2005 2:19 AM Microsoft Corporation c:\windows\system32\wininet.dll

wzcsapi 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 41.00 KB (41,984 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\wzcsapi.dll

netcfgx 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 763.00 KB (781,312 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\netcfgx.dll

rasmans 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 176.00 KB (180,224 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\rasmans.dll

rastapi 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 62.00 KB (63,488 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\rastapi.dll

rasppp 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 205.00 KB (209,920 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\rasppp.dll

ntlsapi 5.2.3790.0 (srv03\_rtm.030324-2048) 8.00 KB (8,192 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\ntlsapi.dll

ipbootp 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 36.00 KB (36,864 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\ipbootp.dll

rasdlg 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 663.00 KB (678,912 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\rasdlg.dll

rasadhlp 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 7.50 KB (7,680 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\rasadhlp.dll

winmr 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 17.00 KB (17,408 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\winmr.dll

spoolsv 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 57.00 KB (58,368 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\spoolsv.exe

spoolss 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 85.00 KB (87,040 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\spoolss.dll

localspl 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 339.00 KB (347,136 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\localspl.dll

cnbjmon 5.2.3790.1224 (dnssrv(skatarl).040514-1058) 46.50 KB (47,616 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\cnbjmon.dll

pjimon 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 15.00 KB  
 (15,360 bytes) 4/14/2005 10:01 AM  
 Microsoft Corporation  
 c:\windows\system32\pjimon.dll

tcpmon 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 47.00 KB  
 (48,128 bytes) 4/14/2005 10:01 AM  
 Microsoft Corporation  
 c:\windows\system32\tcpmon.dll

wsnmp32 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 43.00 KB  
 (44,032 bytes) 4/14/2005 10:00 AM  
 Microsoft Corporation  
 c:\windows\system32\wsnmp32.dll

tcpmib 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 17.50 KB  
 (17,920 bytes) 4/14/2005 10:01 AM  
 Microsoft Corporation  
 c:\windows\system32\tcpmib.dll

wsock32 5.2.3790.0 (srv03\_rtm.030324-2048)  
 22.00 KB (22,528 bytes)  
 4/12/2005 1:43 PM  
 Microsoft Corporation  
 c:\windows\system32\wsock32.dll

mgmtapi 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 15.50 KB  
 (15,872 bytes) 4/12/2005 1:42 PM  
 Microsoft Corporation  
 c:\windows\system32\mgmtapi.dll

snmpapi 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 19.50 KB  
 (19,968 bytes) 4/14/2005 10:01 AM  
 Microsoft Corporation  
 c:\windows\system32\snmpapi.dll

usbmon 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 17.00 KB  
 (17,408 bytes) 4/14/2005 10:01 AM  
 Microsoft Corporation  
 c:\windows\system32\usbmon.dll

wshqos 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 24.00 KB  
 (24,576 bytes) 4/14/2005 10:00 AM  
 Microsoft Corporation  
 c:\windows\system32\wshqos.dll

win32spl 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 100.50  
 KB (102,912 bytes) 4/12/2005 1:43 PM  
 Microsoft Corporation  
 c:\windows\system32\win32spl.dll

inetpp 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 75.00 KB  
 (76,800 bytes) 4/14/2005 10:01 AM  
 Microsoft Corporation  
 c:\windows\system32\inetpp.dll

icmp 5.2.3790.0 (srv03\_rtm.030324-2048)  
 4.50 KB (4,608 bytes)  
 4/12/2005 1:42 PM  
 Microsoft Corporation  
 c:\windows\system32\icmp.dll

aclient 6.5.241 4.75 MB (4,984,908  
 bytes) 5/3/2005 8:59 AM Altiris,  
 Inc.  
 c:\program  
 files\altiris\acclient\aclient.dll  
 (srv03\_sp1\_rtm.050324-1447) 274.50  
 KB (281,088 bytes) 4/12/2005 1:42 PM  
 Microsoft Corporation  
 c:\windows\system32\comdlg32.dll

riched32 5.2.3790.0 (srv03\_rtm.030324-2048)  
 3.50 KB (3,584 bytes)  
 4/12/2005 1:42 PM  
 Microsoft Corporation  
 c:\windows\system32\riched32.dll

riched20 5.31.23.122439.00 KB (449,536  
 bytes) 4/14/2005 10:01 AM  
 Microsoft Corporation  
 c:\windows\system32\riched20.dll

ersvc 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 24.00 KB  
 (24,576 bytes) 4/14/2005 10:01 AM  
 Microsoft Corporation  
 c:\windows\system32\ersvc.dll

inetinfo 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 14.00 KB  
 (14,336 bytes) 4/14/2005 10:02 AM  
 Microsoft Corporation  
 c:\windows\system32\inetinfo.dll

fo.exe 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 164.00  
 KB (167,936 bytes) 4/14/2005 10:03 AM  
 Microsoft Corporation  
 c:\windows\system32\inetres\fo.exe

iisutil 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 164.00  
 KB (167,936 bytes) 4/14/2005 10:03 AM  
 Microsoft Corporation  
 c:\windows\system32\inetres\iisutil.dll

rpcref 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 4.00 KB  
 (4,096 bytes) 4/14/2005 10:01 AM  
 Microsoft Corporation  
 c:\windows\system32\inetres\rpcref.dll

iisrtl 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 138.50  
 KB (141,824 bytes) 4/14/2005 10:02 AM  
 Microsoft Corporation  
 c:\windows\system32\inetres\iisrtl.dll

iisadmin 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 21.00 KB  
 (21,504 bytes) 4/14/2005 10:01 AM  
 Microsoft Corporation  
 c:\windows\system32\inetres\iisadmin.dll

min.dll 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 62.50 KB  
 (64,000 bytes) 4/14/2005 10:01 AM  
 Microsoft Corporation  
 c:\windows\system32\inetres\min.dll

admwprox 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 47.00 KB  
 (48,128 bytes) 4/14/2005 10:02 AM  
 Microsoft Corporation  
 c:\windows\system32\inetres\admwprox.dll

iiscfg 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 1.08 MB  
 (1,133,056 bytes) 4/14/2005 10:02 AM  
 Microsoft Corporation  
 c:\windows\system32\inetres\iiscfg.dll

metadata 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 229.00  
 KB (234,496 bytes) 4/14/2005 10:02 AM  
 Microsoft Corporation  
 c:\windows\system32\inetres\metadata.dll

svcxext 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 43.50 KB  
 (44,544 bytes) 4/14/2005 10:02 AM  
 Microsoft Corporation  
 c:\windows\system32\inetres\svcxext.dll

security 5.2.3790.0 (srv03\_rtm.030324-2048)  
 5.50 KB (5,632 bytes)  
 4/12/2005 1:43 PM  
 Microsoft Corporation  
 c:\windows\system32\security.dll

iismap 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 58.50 KB  
 (59,904 bytes) 4/14/2005 10:03 AM  
 Microsoft Corporation  
 c:\windows\system32\iismap.dll

wamreg 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 54.50 KB  
 (55,808 bytes) 4/14/2005 10:02 AM  
 Microsoft Corporation  
 c:\windows\system32\inetres\wamreg.dll

iisw3adm 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 211.00  
 KB (216,064 bytes) 4/14/2005 10:02 AM  
 Microsoft Corporation  
 c:\windows\system32\inetres\iisw3adm.dll

w3cache 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 19.00 KB  
 (19,456 bytes) 4/14/2005 10:02 AM  
 Microsoft Corporation  
 c:\windows\system32\inetres\w3cache.dll

w3tp 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 13.00 KB  
 (13,312 bytes) 4/14/2005 10:02 AM  
 Microsoft Corporation  
 c:\windows\system32\inetres\w3tp.dll

lonsint 6.0.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 13.00 KB  
 (13,312 bytes) 4/14/2005 10:02 AM  
 Microsoft Corporation  
 c:\windows\system32\inetres\lonsint.dll

termsrv 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 239.00  
 KB (244,736 bytes) 4/14/2005 10:01 AM  
 Microsoft Corporation  
 c:\windows\system32\termsrv.dll

icaapi 5.2.3790.1830  
 (srv03\_sp1\_rtm.050324-1447) 12.50 KB  
 (12,800 bytes) 4/14/2005 10:01 AM  
 Microsoft Corporation  
 c:\windows\system32\icaapi.dll

mstlsapi 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 116.00  
KB (118,784 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\mstlsapi.dll

rdpwsx 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 101.63  
KB (104,072 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\rdpwsx.dll

rdpclip 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 68.00 KB  
(69,632 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\rdpclip.exe

urlmon 6.00.3790.2564  
(srv03\_sp1\_gdr.051104-1524) 675.00  
KB (691,200 bytes) 11/7/2005 2:19 AM  
Microsoft Corporation  
c:\windows\system32\urlmon.dll

explorer 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 1.00 MB  
(1,050,624 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\explorer.exe

browseui 6.00.3790.2564  
(srv03\_sp1\_gdr.051104-1524) 1,012.00  
KB (1,036,288 bytes) 11/7/2005 2:19 AM  
Microsoft Corporation  
c:\windows\system32\browseui.dll

shdocvw 6.00.3790.2580  
(srv03\_sp1\_gdr.051130-1605) 1.44 MB  
(1,513,472 bytes) 12/1/2005 7:13 AM  
Microsoft Corporation  
c:\windows\system32\shdocvw.dll

themeui 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 377.50  
KB (386,560 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\themeui.dll

msimg32 5.2.3790.0 (srv03\_rtm.030324-  
2048) 4.50 KB (4,608 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\msimg32.dll

actxprxy 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 96.50 KB  
(98,816 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\actxprxy.dll

linkinfo 5.2.3790.2521  
(srv03\_sp1\_gdr.050831-1529) 20.00 KB  
(20,480 bytes) 8/31/2005 7:18 PM  
Microsoft Corporation  
c:\windows\system32\linkinfo.dll

ntshrui 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 140.00  
KB (143,360 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\ntshrui.dll

msi 3.1.4000.2435 2.76 MB  
(2,890,240 bytes) 2/1/2006 11:03 AM  
Microsoft Corporation  
c:\windows\system32\msi.dll

webcheck 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 272.50  
KB (279,040 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\webcheck.dll

stobject 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 120.50  
KB (123,392 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\stobject.dll

batmeter 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 31.50 KB  
(32,256 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\batmeter.dll

powrprof 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 16.50 KB  
(16,896 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\powrprof.dll

drprov 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 14.00 KB  
(14,336 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\drprov.dll

ntlanman 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 43.50 KB  
(44,544 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\ntlanman.dll

netui0 5.2.3790.0 (srv03\_rtm.030324-  
2048) 75.50 KB (77,312 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\netui0.dll

netui1 5.2.3790.0 (srv03\_rtm.030324-  
2048) 184.00 KB (188,416 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\netui1.dll

davclnt 5.2.3790.0 (srv03\_rtm.030324-  
2048) 23.50 KB (24,064 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\davclnt.dll

shdoclc 6.00.3790.0 (srv03\_rtm.030324-  
2048) 588.50 KB (602,624 bytes)  
4/12/2005 1:43 PM  
Microsoft Corporation  
c:\windows\system32\shdoclc.dll

acntusr 6, 5, 241 180.00 KB (184,320  
bytes) 5/3/2005 8:59 AM  
c:\program  
files\altiris\acient\acntusr.exe

tapisrv 5.2.3790.2483  
(srv03\_sp1\_gdr.050707-1651) 251.50  
KB (257,536 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\tapisrv.dll

unimdm 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 203.00  
KB (207,872 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\unimdm.tsp

uniplat 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 13.00 KB  
(13,312 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\uniplat.dll

kmddsp 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 34.50 KB  
(35,328 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\kmddsp.tsp

ndptsp 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 47.00 KB  
(48,128 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\ndptsp.tsp

ipconf 5.2.3790.0 (srv03\_rtm.030324-  
2048) 16.50 KB (16,896 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\ipconf.tsp

h323 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 259.50  
KB (265,728 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\h323.tsp

hidphone 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 29.50 KB  
(30,208 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\hidphone.tsp

hid 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 18.50 KB  
(18,944 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\hid.dll

w3wp 6.0.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 7.00 KB  
(7,168 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\inet\w3wp  
.exe

w3core 6.0.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 340.50  
KB (348,672 bytes) 4/14/2005 10:02 AM  
Microsoft Corporation  
c:\windows\system32\inet\w3cor  
e.dll

w3comlog 6.0.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 10.50 KB  
(10,752 bytes) 4/14/2005 10:02 AM  
Microsoft Corporation  
c:\windows\system32\inet\w3co  
mlog.dll

w3dt 6.0.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 38.50 KB  
(39,424 bytes) 4/14/2005 10:02 AM  
Microsoft Corporation  
c:\windows\system32\inet\w3dt.  
dll

iisres 6.0.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 120.00  
KB (122,880 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\inetrv\iisres.

dll

aspnet\_filter 2.0.50727.42 (RTM.050727-4200)  
10.50 KB (10,752 bytes)  
9/23/2005 8:28 AM  
Microsoft Corporation  
c:\windows\microsoft.net\framework  
k\v2.0.50727\aspnet\_filter.dll

msvcr80 8.00.50727.42 612.00  
KB (626,688 bytes) 9/23/2005 8:29 AM  
Microsoft Corporation  
c:\windows\winsxs\x86\_microsoft.v  
c80.crt\_1fc8b3b9a1e18e3b\_8.0.50727.42\_x-  
ww\_0de06acd\msvcr80.dll

w3isapi 6.0.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 61.00 KB  
(62,464 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\inetrv\w3isa

pi.dll

gzip 6.0.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 25.00 KB  
(25,600 bytes) 4/14/2005 10:02 AM  
Microsoft Corporation  
c:\windows\system32\inetrv\gzip.d

ll

"\?c:\inetpub\wwwroot\tpcc.dll"

"\?c:\inetpub\wwwroot\tpcc.dll"

msvcr70 7.00.9466.0 336.00 KB (344,064  
bytes) 4/14/2005 12:11 PM  
Microsoft Corporation  
c:\windows\system32\msvcr70.dll

tpcc\_com Not Available 10.50 KB (10,752  
bytes) 2/6/2006 1:12 PM Not  
Available c:\inetpub\wwwroot\tpcc\_com.dll

tpcc\_odbc Not Available 20.00 KB (20,480  
bytes) 2/6/2006 1:12 PM Not  
Available c:\inetpub\wwwroot\tpcc\_odbc.dll

odbc32 3.526.1830.0  
(srv03\_sp1\_rtm.050324-1447) 240.00  
KB (245,760 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\odbc32.dll

odbcint 3.526.1830.0  
(srv03\_sp1\_rtm.050324-1447) 92.00 KB  
(94,208 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\odbcint.dll

sqlsrv32 2000.086.1830.00  
(srv03\_sp1\_rtm.050324-1447) 436.00  
KB (446,464 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\sqlsrv32.dll

sqlunirl 2000.080.0728.00 176.56  
KB (180,800 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\sqlunirl.dll

sqlsrv32 2000.086.1830.00  
(srv03\_sp1\_rtm.050324-1447) 88.00 KB  
(90,112 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\sqlsrv32.rll

odbc32 3.526.1830.0  
(srv03\_sp1\_rtm.050324-1447) 100.00  
KB (102,400 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\odbc32.dll

dbnetlib 2000.086.1830  
(srv03\_sp1\_rtm.050324-1447) 112.00  
KB (114,688 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\dbnetlib.dll

tpcc\_com\_all 1, 0, 0, 1 104.00 KB (106,496  
bytes) 2/6/2006 1:12 PM  
c:\inetpub\wwwroot\tpcc\_com\_all.d

ll

dllhost 5.2.3790.0 (srv03\_rtm.030324-  
2048) 5.50 KB (5,632 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\dllhost.exe

bxflg 2001.12.4720.2492  
(srv03\_sp1\_gdr.050720-1521) 96.50 KB  
(98,816 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\bxflg.dll

xolehlp 2001.12.4720.1830  
(srv03\_sp1\_rtm.050324-1447) 10.50 KB  
(10,752 bytes) 4/14/2005 10:00 AM  
Microsoft Corporation  
c:\windows\system32\xolehlp.dll

msdtcprx 2001.12.4720.2492  
(srv03\_sp1\_gdr.050720-1521) 455.50  
KB (466,432 bytes) 7/20/2005 8:24 PM  
Microsoft Corporation  
c:\windows\system32\msdtcprx.dll

mtxclu 2001.12.4720.2492  
(srv03\_sp1\_gdr.050720-1521) 77.00 KB  
(78,848 bytes) 7/20/2005 8:24 PM  
Microsoft Corporation  
c:\windows\system32\mtxclu.dll

resutils 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 63.50 KB  
(65,024 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\resutils.dll

catsrv 2001.12.4720.2492  
(srv03\_sp1\_gdr.050720-1521) 268.00  
KB (274,432 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\catsrv.dll

clbcatex 2001.12.4720.2492  
(srv03\_sp1\_gdr.050720-1521) 102.50  
KB (104,960 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\clbcatex.dll

mmc 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 829.00  
KB (848,896 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\mmc.exe

mfc42u 6.06.8063.0 1.11 MB (1,163,776  
bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\mfc42u.dll

mmcbase 5.2.3790.0 (srv03\_rtm.030324-  
2048) 70.50 KB (72,192 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\mmcbase.dll

oleacc 4.2.5406.0 (srv03\_rtm.030324-  
2048) 171.00 KB (175,104 bytes)  
4/12/2005 1:42 PM  
Microsoft Corporation  
c:\windows\system32\oleacc.dll

mmcndmgr 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 1.17 MB  
(1,229,824 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\mmcndmgr.d

ll

mlang 6.00.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 577.50  
KB (591,360 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\mlang.dll

mycomput 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 99.00 KB  
(101,376 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\mycomput.dll

ntmsmgr 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 503.00  
KB (515,072 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\ntmsmgr.dll

ntmsapi 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 44.00 KB  
(45,056 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\ntmsapi.dll

dfrgsnap 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 38.50 KB  
(39,424 bytes) 4/14/2005 10:01 AM  
Microsoft Corp. and Executive  
Software International, Inc.  
c:\windows\system32\dfrgsnap.dll

dfrgres 5.2.3790.0 (srv03\_rtm.030324-  
2048) 50.50 KB (51,712 bytes)  
4/12/2005 1:42 PM  
Microsoft Corp. and Executive  
Software International, Inc.  
c:\windows\system32\dfrgres.dll

dmdskmgr 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 170.00  
KB (174,080 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\dmdskmgr.dll

dmutil 5.2.3790.1830  
(srv03\_sp1\_rtm.050324-1447) 51.50 KB  
(52,736 bytes) 4/14/2005 10:01 AM  
Microsoft Corporation  
c:\windows\system32\dmutil.dll

dmdskres 5.2.3790.0 (srv03\_rtm.030324-2048) 115.50 KB (118,272 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\dmdskres.dll

els 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 196.50 KB (201,216 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\els.dll

filemgmt 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 335.50 KB (343,552 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\filemgmt.dll

localesec 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 228.50 KB (233,984 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\localesec.dll

adsnt 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 272.50 KB (279,040 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\adsnt.dll

smlogcfg 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 369.00 KB (377,856 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\smlogcfg.dll

pdh 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 298.50 KB (305,664 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\pdh.dll

odbcbc 2000.086.1830.00 (srv03\_sp1\_rtm.050324-1447) 24.00 KB (24,576 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\odbcbc.dll

devmgr 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 284.50 KB (291,328 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\devmgr.dll

helpctr 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 778.00 KB (796,672 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\pchealth\helpctr\binaries\helpctr.exe

hcappres 5.2.3790.0 (srv03\_rtm.030324-2048) 6.50 KB (6,656 bytes) 4/13/2005 3:14 PM Microsoft Corporation c:\windows\pchealth\helpctr\binaries\hcappres.dll

its 5.2.3790.2427 (srv03\_sp1\_rtm.050421-1629) 134.00 KB (137,216 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\its.dll

pchshell 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 104.50 KB (107,008 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\pchealth\helpctr\binaries\pchshell.dll

mshtml 6.00.3790.2577 (srv03\_sp1\_rtm.051123-1244) 2.97 MB (3,112,448 bytes) 11/23/2005 5:06 PM Microsoft Corporation c:\windows\system32\mshtml.dll

msls31 3.10.349.0 142.00 KB (145,408 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\msls31.dll

msimtf 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 156.00 KB (159,744 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\msimtf.dll

msctf 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 311.00 KB (318,464 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\msctf.dll

jscrip 5.6.0.8827 448.00 KB (458,752 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\jscrip.dll

imm32 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 108.00 KB (110,592 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\imm32.dll

mshtml 6.00.3790.1830 (srv03\_sp1\_rtm.050324-1447) 454.50 KB (465,408 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\mshtml.dll

vbscript 5.6.0.8827 392.00 KB (401,408 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\vbscript.dll

msinfo 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 376.00 KB (385,024 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\pchealth\helpctr\binaries\msinfo.dll

helpsvc 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 745.00 KB (762,880 bytes) 4/14/2005 10:01 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        |
| Altiris Client Service                     | AClient      | Running       |
|  | Auto         | Own Process   |
|  | c:\program   |               |
| files\altiris\aclient\aclient.exe -service | Normal       | LocalSystem 0 |

Application Experience Lookup Service AeLookupSvcRunning Auto Share Process c:\windows\system32\svchost.exe -

k netsvcs Normal LocalSystem 0

Alerter Alerter 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 Share Process Manual c:\windows\system32\svchost.exe -

k netsvcs Normal LocalSystem 0

ASP.NET State Service aspnet\_state Stopped Manual Own Process c:\windows\microsoft.net\framework\v2.0.50727\aspnet\_state.exe Normal NT AUTHORITY\NetworkService 0

Windows Audio AudioSrv Running Auto 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 netsvcs Normal LocalSystem 0

Computer Browser Browser Running Auto Share Process c:\windows\system32\svchost.exe -

k netsvcs Normal LocalSystem 0

Indexing Service CSvc 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

.NET Runtime Optimization Service v2.0.50727\_X86 clr\_optimization\_v2.0.50727\_32 Stopped Manual Own Process c:\windows\microsoft.net\framework\v2.0.50727\mscorsvw.exe Ignore LocalSystem 0

COM+ System Application COMSysApp Running Manual Own Process c:\windows\system32\dllhost.exe /processid:{02d4b3f1-fd88-11d1-960d-00805fc79235} Normal LocalSystem 0

Cryptographic Services CryptSvc Running Auto Share Process c:\windows\system32\svchost.exe -

k netsvcs Normal LocalSystem 0

DCOM Server Process Launcher  
 DcomLaunch Running Auto  
 Share Process  
 c:\windows\system32\svchost.exe -  
 k dcomlaunch Normal LocalSystem 0

Distributed File System Dfs Stopped  
 Manual 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 Stopped 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 networkservice 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 Running  
 Auto 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  
 Auto Share Process  
 c:\windows\system32\svchost.exe -  
 k networkservice Normal LocalSystem 0

Help and Support helpsvc Running  
 Auto Share Process  
 c:\windows\system32\svchost.exe -  
 k networkservice Normal LocalSystem 0

Human Interface Device Access HidServ  
 Stopped Disabled Share  
 Process c:\windows\system32\svchost.exe -  
 k networkservice Normal LocalSystem 0

HTTP SSL HTTPFilter Running Manual  
 Share Process  
 c:\windows\system32\sass.exe  
 Normal LocalSystem 0

IIS Admin Service IISADMIN Running  
 Auto Share Process  
 c:\windows\system32\inet\inetin  
 fo.exe Normal LocalSystem 0

IMAPI CD-Burning COM Service  
 ImapiService Stopped Disabled  
 Own Process  
 c:\windows\system32\imapi.exe  
 Normal LocalSystem 0

Intersite Messaging IsmServ Stopped  
 Disabled Own Process  
 c:\windows\system32\ismerv.exe  
 Normal LocalSystem 0

Kerberos Key Distribution Center kdc  
 Stopped Disabled Share  
 Process c:\windows\system32\sass.exe  
 Normal LocalSystem 0

Server lanmanserver Running Auto  
 Share Process  
 c:\windows\system32\svchost.exe -  
 k networkservice Normal LocalSystem 0

Workstation lanmanworkstation Running  
 Auto Share Process  
 c:\windows\system32\svchost.exe -  
 k networkservice Normal LocalSystem 0

License Logging LicenseService  
 Stopped Disabled Own  
 Process c:\windows\system32\lssrv.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 networkservice Normal LocalSystem 0

NetMeeting Remote Desktop Sharing  
 mnmsrv Stopped Disabled  
 Own Process  
 c:\windows\system32\mnmsrv.exe  
 Normal LocalSystem 0

Distributed Transaction Coordinator MSDTC  
 Running Auto Own  
 Process c:\windows\system32\msdtc.exe  
 Normal NT  
 AUTHORITY\NetworkService 1

Windows Installer MSIServer Stopped  
 Manual Share Process  
 c:\windows\system32\msiexec.exe  
 /v Normal LocalSystem 0

Network DDE NetDDE Stopped Disabled  
 Share Process  
 c:\windows\system32\netdde.exe  
 Normal LocalSystem 0

Network DDE DSDM NetDDEdsdm Stopped  
 Disabled Share Process  
 c:\windows\system32\netdde.exe  
 Normal LocalSystem 0

Net Logon Netlogon Stopped Manual  
 Share Process  
 c:\windows\system32\sass.exe  
 Normal LocalSystem 0

Network Connections Netman Running  
 Manual Share Process  
 c:\windows\system32\svchost.exe -  
 k networkservice Normal LocalSystem 0

Network Location Awareness (NLA) Nla  
 Running Manual Share  
 Process c:\windows\system32\svchost.exe -  
 k networkservice Normal LocalSystem 0

File Replication NTFRs Stopped  
 Manual Own Process  
 c:\windows\system32\ntfrs.exe  
 Ignore LocalSystem 0

NT LM Security Support Provider NtLmSsp  
 Stopped Manual Share  
 Process c:\windows\system32\sass.exe  
 Normal LocalSystem 0

Removable Storage NtmsSvc Stopped  
 Manual Share Process  
 c:\windows\system32\svchost.exe -  
 k networkservice Normal LocalSystem 0

Plug and Play PlugPlay Running Auto  
 Share Process  
 c:\windows\system32\services.exe  
 Normal LocalSystem 0

IPSEC Services PolicyAgent Running  
 Auto Share Process  
 c:\windows\system32\sass.exe  
 Normal LocalSystem 0

Protected Storage ProtectedStorage  
 Running Auto Share  
 Process c:\windows\system32\sass.exe  
 Normal LocalSystem 0

Remote Access Auto Connection Manager  
 RasAuto Stopped Manual  
 Share Process  
 c:\windows\system32\svchost.exe -  
 k networkservice Normal LocalSystem 0

Remote Access Connection Manager RasMan  
 Running Manual Share  
 Process c:\windows\system32\svchost.exe -  
 k networkservice Normal LocalSystem 0

Remote Desktop Help Session Manager  
 RDSessMgr 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 networkservice Normal LocalSystem 0

Remote Registry RemoteRegistry  
 Running Auto Share  
 Process c:\windows\system32\svchost.exe -  
 k regsvc Normal NT  
 AUTHORITY\LocalService 0

Remote Command Service RMSYS Running  
 Auto Own Process  
 "c:\program  
 files\benchcraft\sys.exe" Normal  
 .\Administrator 0

Remote Procedure Call (RPC) Locator  
 RplLocator 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 rpcss Normal NT  
 Authority\NetworkService 0

Resultant Set of Policy Provider  
 RSoPProv Stopped Manual  
 Share Process  
 c:\windows\system32\rsopprov.exe  
 Normal LocalSystem 0

Special Administration Console Helper sacsvr  
 Stopped Manual Share  
 Process c:\windows\system32\svchost.exe -  
 k netsvcs Normal LocalSystem 0

Security Accounts ManagerSamSs Running  
 Auto Share Process  
 c:\windows\system32\lsass.exe  
 Normal LocalSystem 0

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 Running  
 Auto 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

Windows Firewall/Internet Connection Sharing (ICS) SharedAccess Stopped  
 Disabled 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 netsvcs Ignore LocalSystem 0

Print Spooler Spooler Running Auto  
 Own Process  
 c:\windows\system32\spoolsv.exe  
 Normal LocalSystem 0

Windows Image Acquisition (WIA) stisvc  
 Stopped Disabled Share  
 Process c:\windows\system32\svchost.exe -  
 k imgsvc Normal NT  
 AUTHORITY\LocalService 0

Microsoft Software Shadow Copy Provider swprv Stopped Manual  
 Own Process  
 c:\windows\system32\svchost.exe -  
 k swprv Normal LocalSystem 0

Performance Logs and Alerts SysmonLog Stopped Manual  
 Own Process  
 c:\windows\system32\smlogsvc.exe  
 Normal NT

Telephone Network Service Running 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 termsvcs Normal LocalSystem 0

Themes Themes Stopped Disabled  
 Share Process  
 c:\windows\system32\svchost.exe -  
 k netsvcs 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 netsvcs Normal LocalSystem 0

Distributed Link Tracking Client TrkWks  
 Running Auto Share  
 Process c:\windows\system32\svchost.exe -  
 k netsvcs Normal LocalSystem 0

Terminal Services Session Directory Tssdis  
 Stopped Disabled Own  
 Process c:\windows\system32\tssdis.exe  
 Normal LocalSystem 0

Windows User Mode Driver Framework UMWdf  
 Stopped Manual Own  
 Process c:\windows\system32\wdfmgr.exe  
 Normal NT  
 AUTHORITY\LocalService 0

Upload Manager uploadmgr Stopped  
 Manual Share Process  
 c:\windows\system32\svchost.exe -  
 k netsvcs Normal LocalSystem 0

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\vds.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 localservice Normal NT  
 AUTHORITY\LocalService 0

World Wide Web Publishing Service W3SVC  
 Running Auto Share  
 Process c:\windows\system32\svchost.exe -  
 k iissvcs Normal LocalSystem 0

WebClient WebClient Stopped Disabled  
 Share Process  
 c:\windows\system32\svchost.exe -  
 k localservice Normal NT  
 AUTHORITY\LocalService 0

Web Client Discovery Service WinHttpAutoProxySvc Stopped  
 Manual Share Process  
 c:\windows\system32\svchost.exe -  
 k localservice Normal NT  
 AUTHORITY\LocalService 0

Windows Management Instrumentation winmgmt Running Auto  
 Share Process  
 c:\windows\system32\svchost.exe -  
 k netsvcs Ignore LocalSystem 0

Portable Media Serial Number Service WmdmPmSN Stopped Manual  
 Share Process  
 c:\windows\system32\svchost.exe -  
 k netsvcs Normal 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\wmiapi  
 srv.exe Normal LocalSystem 0

Automatic Updates wuauclt Running  
 Auto Share Process  
 c:\windows\system32\svchost.exe -  
 k netsvcs Normal LocalSystem 0

Wireless Configuration WZCSVC Running  
 Auto Share Process  
 c:\windows\system32\svchost.exe -  
 k netsvcs Normal LocalSystem 0

Network Provisioning Service xmlprov  
 Stopped Manual Share  
 Process c:\windows\system32\svchost.exe -  
 k netsvcs Normal LocalSystem 0

[Program Groups]

| Group Name                 | Name                                   | User Name    | Default   |
|----------------------------|--|--------------|-----------|
| Accessories                | Default User:Accessories               | Default User | Default   |
| Accessories\Accessibility  | Default User:Accessories\Accessibility | Default User | Default   |
| Accessories\Entertainment  | Default User:Accessories\Entertainment | Default User | Default   |
| Startup                    | Default User:Startup                   | Default User | Default   |
| Accessories                | All Users:Accessories                  | All Users    | All Users |
| Accessories\Accessibility  | All Users:Accessories\Accessibility    | All Users    | All Users |
| Accessories\Communications | All Users:Accessories\Communications   | All Users    | All Users |
| Accessories\Entertainment  | All Users:Accessories\Entertainment    | All Users    | All Users |

Accessories\System Tools All Users:Accessories\System Tools All Users

Administrative Tools All Users:Administrative Tools All Users Kubelka All Users:Kubelka All Users

Microsoft Network Monitor All Users:Microsoft Network Monitor All Users  
 Microsoft SQL Server All Users:Microsoft SQL Server All Users  
 Microsoft Visual Studio .NET 2003 All Users:Microsoft Visual Studio .NET 2003 All Users

Microsoft Visual Studio .NET 2003\Visual Studio .NET Tools All Users:Microsoft Visual Studio .NET 2003\Visual Studio .NET Tools All Users

Startup All Users:Startup All Users

Sysinternals PsTools All Users:Sysinternals PsTools 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 DL18\Administrator:Accessories DL18\Administrator  
 Accessories\Accessibility DL18\Administrator:Accessories\Accessibility DL18\Administrator  
 Accessories\Communications DL18\Administrator:Accessories\Communications DL18\Administrator  
 Accessories\Communications\HyperTerminal DL18\Administrator:Accessories\Communications\HyperTerminal DL18\Administrator  
 Accessories\Entertainment DL18\Administrator:Accessories\Entertainment DL18\Administrator  
 Administrative Tools DL18\Administrator:Administrative Tools DL18\Administrator  
 Benchcraft DL18\Administrator:Benchcraft DL18\Administrator  
 QLogic Corporation DL18\Administrator:QLogic Corporation DL18\Administrator  
 QLogic Corporation\SANblade Control VIX DL18\Administrator:QLogic Corporation\SANblade Control VIX DL18\Administrator  
 Startup DL18\Administrator:Startup DL18\Administrator

[Startup Programs]

| Program | Command     | User Name           | Location |
|---------|-------------|---------------------|----------|
| desktop | desktop.ini | NT AUTHORITY\SYSTEM | Startup  |
| desktop | desktop.ini | DL18\Administrator  | Startup  |
| desktop | desktop.ini | .DEFAULT            | Startup  |

desktop desktop.ini All Users Common  
 c:\program files\altiris\client\acntusr.exe All Users  
 HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

[OLE Registration]

Object Local Server  
 Sound (OLE2) sndrec32.exe  
 Media Clip mplay32.exe  
 Video Clip mplay32.exe /avi  
 MIDI Sequence mplay32.exe /mid

Sound Not Available  
 Media Clip Not Available  
 WordPad Document "%programfiles%\windows\n\t\accessories\wordpad.exe"  
 Windows Media Services DRM Storage object Not Available

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  | 6.0.3790.1830                      |         |
| Build  | 63790.1830                         |         |
| 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                 |
|--------------|---------------|----------|----------------------|
| actxprxy.dll | 6.0.3790.1830 | 97 KB    | 3/24/2005 5:55:26 PM |
| advpack.dll  | 6.0.3790.1830 | 98 KB    | 3/24/2005 5:55:28 PM |
| asctrls.ocx  | 6.0.3790.0    | 90 KB    | 3/25/2003 5:00:00 AM |
| browsecl.dll | 6.0.3790.0    | 62 KB    | 3/25/2003 5:00:00 AM |
| browseui.dll | 6.0.3790.2564 | 1,012 KB | 11/7/2005 3:19:28 AM |

|               |                |               |                       |
|---------------|----------------|---------------|-----------------------|
| cdfview.dll   | 6.0.3790.1830  | 149 KB        | 3/24/2005 5:56:32 PM  |
| comctl32.dll  | 6.0.3790.1830  | 585 KB        | 3/24/2005 5:57:56 PM  |
| dxtrans.dll   | 6.3.3790.2564  | 208 KB        | 11/7/2005 3:19:28 AM  |
| dxtransf.dll  | 6.3.3790.1830  | 355 KB        | 3/24/2005 6:00:58 PM  |
| iecont.dll    | <File Missing> | Not Available | Not Available         |
| iecontlc.dll  | <File Missing> | Not Available | Not Available         |
| iedkcs32.dll  | 16.0.3790.1830 | 324 KB        | 3/24/2005 6:04:58 PM  |
| iepeers.dll   | 6.0.3790.2564  | 248 KB        | 11/7/2005 3:19:28 AM  |
| iesetup.dll   | 6.0.3790.1830  | 61 KB         | 3/24/2005 6:04:58 PM  |
| ieuinit.inf   | Not Available  | 24 KB         | 3/24/2005 6:04:58 PM  |
| Available     |                |               | Not Available         |
| ieexplore.exe | 6.0.3790.1830  | 92 KB         | 3/24/2005 6:04:58 PM  |
| imgutil.dll   | 6.0.3790.1830  | 38 KB         | 3/24/2005 6:05:04 PM  |
| inetctl.cpl   | 6.0.3790.1830  | 358 KB        | 3/24/2005 6:05:06 PM  |
| inetctlc.dll  | 6.0.3790.0     | 109 KB        | 3/25/2003 5:00:00 AM  |
| inseng.dll    | 6.0.3790.1830  | 94 KB         | 3/24/2005 6:05:06 PM  |
| mlang.dll     | 6.0.3790.1830  | 578 KB        | 3/24/2005 6:07:20 PM  |
| msencode.dll  | 2002.10.4.0    | 112 KB        | 3/25/2003 5:00:00 AM  |
| mshta.exe     | 6.0.3790.1830  | 30 KB         | 3/24/2005 6:07:26 PM  |
| mshtml.dll    | 6.0.3790.2577  | 3,040 KB      | 11/23/2005 6:06:18 PM |



|              |   |          |
|--------------|---|----------|
| mshhtml.tlb  | 6.0.3790.1830<br>3/24/2005 6:07:26 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 1,320 KB |
| mshhtml.dll  | 6.0.3790.1830<br>3/24/2005 6:07:26 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 455 KB   |
| mshhtml.dll  | 6.0.3790.1830<br>3/24/2005 6:07:26 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 56 KB    |
| msident.dll  | 6.0.3790.1830<br>3/24/2005 6:07:28 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 48 KB    |
| msidntld.dll | 6.0.3790.0 15 KB<br>3/25/2003 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation  |          |
| msieftp.dll  | 6.0.3790.1830<br>3/24/2005 6:07:28 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 244 KB   |
| msrating.dll | 6.0.3790.1830<br>3/24/2005 6:07:36 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 144 KB   |
| mstime.dll   | 6.0.3790.2564<br>11/7/2005 3:19:29 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 524 KB   |
| occache.dll  | 6.0.3790.1830<br>3/24/2005 6:08:34 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 94 KB    |
| proctexe.ocx | 6.3.3790.1830<br>3/24/2005 6:12:26 PM<br>C:\WINDOWS\system32<br>Intel Corporation         | 83 KB    |
| sendmail.dll | 6.0.3790.1830<br>3/24/2005 6:13:36 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 56 KB    |
| shdoclc.dll  | 6.0.3790.0 589 KB<br>3/25/2003 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation |          |
| shdocvw.dll  | 6.0.3790.2580<br>12/1/2005 8:13:55 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 1,478 KB |
| shfolder.dll | 6.0.3790.1830<br>3/24/2005 6:13:36 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 25 KB    |
| shlwapi.dll  | 6.0.3790.2564<br>11/7/2005 3:19:29 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 314 KB   |
| tdc.ocx      | 1.3.0.3130 58 KB<br>3/25/2003 5:00:00 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation  |          |
| url.dll      | 6.0.3790.1830<br>3/24/2005 6:26:12 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 37 KB    |
| urlmon.dll   | 6.0.3790.2564<br>11/7/2005 3:19:30 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation     | 675 KB   |

|              |   |        |
|--------------|---|--------|
| webcheck.dll | 6.0.3790.1830<br>3/24/2005 6:26:16 PM<br>C:\WINDOWS\system32<br>Microsoft Corporation | 273 KB |
| wininet.dll  | 6.0.3790.2564<br>11/7/2005 3:19:30 AM<br>C:\WINDOWS\system32<br>Microsoft Corporation | 647 KB |

[Connectivity]

|                       |            |
|-----------------------|------------|
| Item                  | Value      |
| Connection Preference | Never dial |

LAN Settings

|                     |             |
|---------------------|-------------|
| AutoConfigProxy     | wininet.dll |
| AutoProxyDetectMode | Disabled    |
| AutoConfigURL       |             |
| Proxy               | Disabled    |
| ProxyServer         | itgproxy:80 |
| 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\SQLTPCKIT\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 |
| WUWebControl Class  | Installed |          |
| http://update.microsoft.com/windowsupdate/v6/V5Controls/en/x86/client/wuweb_siste.cab?1138820495434 |           |          |
| {9F1C11AA-197B-4942-BA54-47A8489BB47F}  |           |          |
| Not Available   |           |          |
| http://v4.windowsupdate.microsoft.com/CAB/x86/unicode/iuctl.CAB?38455.6793981481                    |           |          |

[Content]

[ Following are sub-categories of this main category ]  
[Summary]

|                 |          |
|-----------------|----------|
| Item            | Value    |
| Content Advisor | Disabled |

[Personal Certificates]

|   |                     |          |
|---|---------------------|----------|
| Issued To                                     | Issued By           | Validity |
|   | Signature Algorithm |          |
| No personal certificate information available |                     |          |

[Other People Certificates]

|           |                     |          |
|-----------|---------------------|----------|
| Issued To | Issued By           | Validity |
|           | Signature Algorithm |          |

No other people certificate information available

[Publishers]

Name  
No publisher information available

[Security]

|                  |                |
|------------------|----------------|
| Zone             | Security Level |
| My Computer      | Custom         |
| Local intranet   | Custom         |
| Trusted sites    | Custom         |
| Internet         | Custom         |
| Restricted sites | Custom         |

**COM+ Settings**

TPCC.AllTxns:

Activation:  
selected Enable Object Pooling  
59 Minimum Pool Size:  
59 Maximum Pool Size:  
60000 Creation Timeout:  
Construction Enable Object  
Activation Enable Just In Time  
Concurrency:  
Concurrency Required

**Microsoft IIS Registry Parameters**

Windows Registry Editor Version 5.00

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]  
"ListenBackLog"=dword:00008ca0  
"PoolThreadLimit"=dword:00003ff8  
"MaxPoolThreads"=dword:00000ffe  
"ThreadTimeout"=dword:00015180

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]  
"Library"="infoctrs.dll"  
"Open"="OpenINFOPerformanceData"  
"Close"="CloseINFOPerformanceData"  
"Collect"="CollectINFOPerformanceData"  
"PerfIniFile"="infoctrs.ini"



```

@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.
vbs SET W3SVC/AppPools/PeriodicRestartMemory
0 >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.
vbs SET W3SVC/AppPools/AppPoolRecycleTime
False >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.
vbs SET
W3SVC/AppPools/AppPoolRecycleRequests False
>> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.
vbs SET
W3SVC/AppPools/AppPoolRecycleSchedule False
>> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.
vbs SET
W3SVC/AppPools/AppPoolRecycleMemory False
>> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.
vbs SET
W3SVC/AppPools/AppPoolRecycleIsapiUnhealthy
False >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.
vbs SET
W3SVC/AppPools/AppPoolRecycleOnDemand
False >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.
vbs SET
W3SVC/AppPools/AppPoolRecycleConfigChange
False >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.
vbs SET
W3SVC/AppPools/AppPoolRecyclePrivateMemory
False >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul

```

```

@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.
vbs SET W3SVC/AppPools/PingingEnabled False
>>
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.
vbs SET W3SVC/AppPools/RapidFailProtection
False >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul

```

### TPCC Application Registry Parameters

Windows Registry Editor Version 5.00

```

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\T
PCC]
"Path"="C:\inetpub\wwwroot\"
"NumberOfDeliveryThreads"=dword:00000008
"MaxConnections"=dword:00008ca0
"MaxPendingDeliveries"=dword:000009c4
"DB_Protocol"="ODBC"
"TxnMonitor"="COM"
"DbServer"="tcp:15.1.104.1,1436"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"="ssdl"
"COM_SinglePool"="YES"
"ConnectDelay"=dword:00000001
"CallNoDuplicatesNewOrder"=dword:00000001

```

NOTE: This is representative of 1 web client. DBServer was varied on every web client to connect to the appropriate SoftNuma node

### TCPIP Registry Parameters

Windows Registry Editor Version 5.00

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControl
Set\Services\Tcpip\Parameters]
"NV Hostname"="dl1"
"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"="dl1"
"SearchList"=""
"UseDomainNameDevolution"=dword:00000001
"EnableICMPRedirect"=dword:00000001
"DeadGWDetectDefault"=dword:00000001

```

```

"DontAddDefaultGatewayDefault"=dword:000000
"EnableSecurityFilters"=dword:00000000
"AllowUnqualifiedQuery"=dword:00000000
"PrioritizeRecordData"=dword:00000001
"NV Domain"=""
"MaxUserPort"=dword:000000ff
"DhcpNameServer"="15.1.101.1"
"DhcpDomain"="hp-perf.net"

```

## RTE Input Parameters

### 3Tier.pro

Profile: SQLSapphire 3 tier  
File Path: D:\SQLSapphire\Runs\Three Tier\7-11-06\SQLSapphire 3 tier.xml  
Version: 5

Number of Engines: 24

|                   |   |
|-------------------|---|
| C:\DRIVER18.log   | Name: DRIVER01<br>Description:<br>Directory:  |
| 10810             | Machine: CL18<br>Parameter Set: Audit<br>Index: 50000000<br>Seed: 73660<br>Configured Users:  |
| 233               | Pipe Name: DRIVER01<br>Connect Rate: 5000<br>Start Rate: 1000<br><br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND: |
| spprefix=sp1      | CPU: 0<br>Additional Options:   |
| C:\DRIVER18-2.log | Name: DRIVER02<br>Description:<br>Directory:  |
| 10810             | Machine: CL18<br>Parameter Set: Audit<br>Index: 75000000<br>Seed: 73660<br>Configured Users:  |
| 233               | Pipe Name: DRIVER02<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND:     |
| spprefix=sp1      | CPU: 1<br>Additional Options:   |
| C:\DRIVER19.log   | Name: DRIVER03<br>Description:<br>Directory:  |

|                   |   |                   |   |                   |   |
|-------------------|---|-------------------|---|-------------------|---|
| 11390             | Machine: CL19<br>Parameter Set: Audit<br>Index: 100000000<br>Seed: 73660<br>Configured Users:                                   | 233               | CLIENT_NURAND:<br>CPU: 1<br>Additional Options:   | 11850             | See figure 7-560<br>Configured Users:   |
| 233               | Pipe Name: DRIVER03<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND: | spprefix=sp3      | Name: DRIVER07<br>Description:<br>Directory:  | 233               | Pipe Name: DRIVER10<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND: |
| spprefix=sp2      | CPU: 0<br>Additional Options:   | C:\DRIVER21.log   | Machine: CL21<br>Parameter Set: Audit<br>Index: 200000000<br>Seed: 73660<br>Configured Users:                                   | spprefix=sp1      | CPU: 1<br>Additional Options:   |
| C:\DRIVER19-2.log | Name: DRIVER04<br>Description:<br>Directory:  | 11850             | Pipe Name: DRIVER07<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND: | C:\DRIVER23.log   | Name: DRIVER11<br>Description:<br>Directory:  |
| 11390             | Machine: CL19<br>Parameter Set: Audit<br>Index: 125000000<br>Seed: 73660<br>Configured Users:                                   | 233               | CPU: 0<br>Additional Options:   | 11850             | Machine: CL23<br>Parameter Set: Audit<br>Index: 300000000<br>Seed: 73660<br>Configured Users:                                   |
| 233               | Pipe Name: DRIVER04<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND: | spprefix=sp4      | Name: DRIVER08<br>Description:<br>Directory:  | 233               | Pipe Name: DRIVER11<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND: |
| spprefix=sp2      | CPU: 1<br>Additional Options:   | C:\DRIVER21-2.log | Machine: CL21<br>Parameter Set: Audit<br>Index: 225000000<br>Seed: 73660<br>Configured Users:                                   | spprefix=sp2      | CPU: 0<br>Additional Options:   |
| C:\DRIVER20.log   | Name: DRIVER05<br>Description:<br>Directory:  | 11850             | Pipe Name: DRIVER08<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND: | C:\DRIVER23-2.log | Name: DRIVER12<br>Description:<br>Directory:  |
| 11850             | Machine: CL20<br>Parameter Set: Audit<br>Index: 150000000<br>Seed: 73660<br>Configured Users:                                   | 233               | CPU: 1<br>Additional Options:   | 11850             | Machine: CL23<br>Parameter Set: Audit<br>Index: 325000000<br>Seed: 73660<br>Configured Users:                                   |
| 233               | Pipe Name: DRIVER05<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND: | spprefix=sp4      | Name: DRIVER09<br>Description:<br>Directory:  | 233               | Pipe Name: DRIVER12<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND: |
| spprefix=sp3      | CPU: 0<br>Additional Options:   | C:\DRIVER22.log   | Machine: CL22<br>Parameter Set: Audit<br>Index: 250000000<br>Seed: 73660<br>Configured Users:                                   | spprefix=sp2      | CPU: 1<br>Additional Options:   |
| C:\DRIVER20-2.log | Name: DRIVER06<br>Description:<br>Directory:  | 11850             | Pipe Name: DRIVER09<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND: | C:\DRIVER24.log   | Name: DRIVER13<br>Description:<br>Directory:  |
| 11850             | Machine: CL20<br>Parameter Set: Audit<br>Index: 175000000<br>Seed: 73660<br>Configured Users:                                   | 233               | CPU: 0<br>Additional Options:   | 11850             | Machine: CL24<br>Parameter Set: Audit<br>Index: 350000000<br>Seed: 73660<br>Configured Users:                                   |
| 233               | Pipe Name: DRIVER06<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND: | spprefix=sp1      | Name: DRIVER10<br>Description:<br>Directory:  | 233               | Pipe Name: DRIVER13<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND: |
|                   | CPU: 0<br>Additional Options:   | C:\DRIVER22-2.log | Machine: CL22<br>Parameter Set: Audit<br>Index: 275000000   |                   | CPU: 0  |

|                   |   |                   |  |                   |   |
|-------------------|---|-------------------|--|-------------------|---|
| spprefix=sp3      | Additional Options:   | 10810             | Configured Users:<br>Pipe Name: DRIVER17<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND: | C:\DRIVER28.log   | Driver28<br>Directory:  |
| C:\DRIVER24-2.log | Name: DRIVER14<br>Description:<br>Directory:<br><br>Machine: CL24<br>Parameter Set: Audit<br>Index: 375000000<br>Seed: 73660<br>Configured Users: | 233               | CPU: 0<br>Additional Options:  | 11850             | Machine: CL28<br>Parameter Set: Audit<br>Index: 575000000<br>Seed: 73660<br>Configured Users:   |
| 11850             | Pipe Name: DRIVER14<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND:                   | C:\DRIVER26-2.log | Name: DRIVER18<br>Description:<br>Directory:<br><br>Machine: CL26<br>Parameter Set: Audit<br>Index: 500000000<br>Seed: 73660<br>Configured Users:    | 233               | Pipe Name: DRIVER21<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND:                   |
| 233               | CPU: 1<br>Additional Options:   | 11390             | Pipe Name: DRIVER18<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND:                      | C:\DRIVER28-2.log | CPU: 0<br>Additional Options:   |
| spprefix=sp3      | Name: DRIVER15<br>Description:<br>Directory:<br><br>Machine: CL25<br>Parameter Set: Audit<br>Index: 400000000<br>Seed: 73660<br>Configured Users: | 233               | CPU: 1<br>Additional Options:  | 11850             | Name: DRIVER22<br>Description:<br>Directory:<br><br>Machine: CL28<br>Parameter Set: Audit<br>Index: 600000000<br>Seed: 73660<br>Configured Users: |
| C:\DRIVER25.log   | Pipe Name: DRIVER15<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND:                   | C:\DRIVER27.log   | Name: DRIVER19<br>Description:<br>Directory:<br><br>Machine: CL27<br>Parameter Set: Audit<br>Index: 525000000<br>Seed: 73660<br>Configured Users:    | 233               | Pipe Name: DRIVER22<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND:                   |
| 9890              | CPU: 0<br>Additional Options:   | 11850             | Pipe Name: DRIVER19<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND:                      | C:\DRIVER29.log   | CPU: 1<br>Additional Options:   |
| spprefix=sp4      | Name: DRIVER16<br>Description:<br>Directory:<br><br>Machine: CL25<br>Parameter Set: Audit<br>Index: 425000000<br>Seed: 73660<br>Configured Users: | 233               | CPU: 0<br>Additional Options:  | 11850             | Name: DRIVER23<br>Description:<br>Directory:<br><br>Machine: CL29<br>Parameter Set: Audit<br>Index: 625000000<br>Seed: 73660<br>Configured Users: |
| C:\DRIVER25-2.log | Pipe Name: DRIVER16<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND:                   | C:\DRIVER27-2.log | Name: DRIVER20<br>Description:<br>Directory:<br><br>Machine: CL27<br>Parameter Set: Audit<br>Index: 550000000<br>Seed: 73660<br>Configured Users:    | 233               | Pipe Name: DRIVER23<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND:                   |
| 9890              | CPU: 1<br>Additional Options:   | 11850             | Pipe Name: DRIVER20<br>Connect Rate: 5000<br>Start Rate: 1000<br>Max. Concurrency: -1<br>Concurrency Rate: 10<br>CLIENT_NURAND:                      | C:\DRIVER29-2.log | CPU: 0<br>Additional Options:   |
| spprefix=sp4      | Name: DRIVER17<br>Description:<br>Directory:<br><br>Machine: CL26<br>Parameter Set: Audit<br>Index: 475000000<br>Seed: 73660                      | 233               | CPU: 1<br>Additional Options:  | 9890              | Name: DRIVER24<br>Description:<br>Directory:<br><br>Machine: CL29<br>Parameter Set: Audit<br>Index: 650000000<br>Seed: 73660<br>Configured Users: |
| C:\DRIVER26.log   |   | spprefix=sp2      |  |                   | Pipe Name: DRIVER24   |

|                     |  |                     |  |                     |  |
|---------------------|--|---------------------|--|---------------------|--|
| 233                 | <p>Connect Rate: 5000<br/>Start Rate: 1000<br/>Max. Concurrency: -1<br/>Concurrency Rate: 10<br/>CLIENT_NURAND:</p> <p>CPU: 1<br/>Additional Options:<br/>spprefix=sp4</p> <p>Number of User groups: 24</p>  | 4440                | <p>Proc Ranger 7902 -<br/>w_id Min Warehouse:<br/>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 11390<br/>District id: 1<br/>Scale Down: No</p> <p>Driver Engine:<br/>IIS Server: DL20<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa<br/>Protocol: HTML<br/>w_id Range: 4441 -</p> <p>w_id Min Warehouse:<br/>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 11850<br/>District id: 1<br/>Scale Down: No</p> <p>Driver Engine:<br/>IIS Server: DL20<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa<br/>Protocol: HTML<br/>w_id Range: 5626 -</p> <p>w_id Min Warehouse:<br/>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 11850<br/>District id: 1<br/>Scale Down: No</p> <p>Driver Engine:<br/>IIS Server: DL21<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa<br/>Protocol: HTML<br/>w_id Range: 6811 -</p> <p>w_id Min Warehouse:<br/>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 11850<br/>District id: 1<br/>Scale Down: No</p> <p>Driver Engine:<br/>IIS Server: DL21<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa</p> | 9180                | <p>Proc Ranger 7996 -<br/>w_id Min Warehouse:<br/>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 11850<br/>District id: 1<br/>Scale Down: No</p> <p>Driver Engine:<br/>IIS Server: DL22<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa<br/>Protocol: HTML<br/>w_id Range: 9181 -</p> <p>w_id Min Warehouse:<br/>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 11850<br/>District id: 1<br/>Scale Down: No</p> <p>Driver Engine:<br/>IIS Server: DL22<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa<br/>Protocol: HTML<br/>w_id Range: 10366 -</p> <p>w_id Min Warehouse:<br/>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 11850<br/>District id: 1<br/>Scale Down: No</p> <p>Driver Engine:<br/>IIS Server: DL23<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa<br/>Protocol: HTML<br/>w_id Range: 11551 -</p> <p>w_id Min Warehouse:<br/>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 11850<br/>District id: 1<br/>Scale Down: No</p> <p>Driver Engine:<br/>IIS Server: DL23<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa</p> |
| DRIVER01            | <p>Driver Engine:<br/>IIS Server: DL18<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa<br/>Protocol: HTML<br/>w_id Range: 1 - 1081<br/>w_id Min Warehouse:</p> <p>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 10810<br/>District id: 1<br/>Scale Down: No</p>  | DRIVER05            | <p>Driver Engine:<br/>IIS Server: DL20<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa<br/>Protocol: HTML<br/>w_id Range: 4441 -</p> <p>w_id Min Warehouse:<br/>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 11850<br/>District id: 1<br/>Scale Down: No</p>  | DRIVER09            | <p>Driver Engine:<br/>IIS Server: DL22<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa<br/>Protocol: HTML<br/>w_id Range: 9181 -</p> <p>w_id Min Warehouse:<br/>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 11850<br/>District id: 1<br/>Scale Down: No</p>  |
| tcp:15.1.104.1,1436 |  | tcp:15.1.104.1,1438 |  | tcp:15.1.104.2,1440 |  |
| 1                   |  | 5625                |  | 10365               |  |
| 27402               |  | 1                   |  | 1                   |  |
| DRIVER02            | <p>Driver Engine:<br/>IIS Server: DL18<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa<br/>Protocol: HTML<br/>w_id Range: 1082 -</p> <p>w_id Min Warehouse:</p> <p>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 10810<br/>District id: 1<br/>Scale Down: No</p> | DRIVER06            | <p>Driver Engine:<br/>IIS Server: DL20<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa<br/>Protocol: HTML<br/>w_id Range: 5626 -</p> <p>w_id Min Warehouse:<br/>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 11850<br/>District id: 1<br/>Scale Down: No</p>  | DRIVER10            | <p>Driver Engine:<br/>IIS Server: DL22<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa<br/>Protocol: HTML<br/>w_id Range: 10366 -</p> <p>w_id Min Warehouse:<br/>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 11850<br/>District id: 1<br/>Scale Down: No</p>   |
| tcp:15.1.104.1,1436 |  | tcp:15.1.104.1,1438 |  | tcp:15.1.104.2,1440 |  |
| 2162                |  | 6810                |  | 11550               |  |
| 1                   |  | 1                   |  | 1                   |  |
| 27402               |  | 27402               |  | 27402               |  |
| DRIVER03            | <p>Driver Engine:<br/>IIS Server: DL19<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa<br/>Protocol: HTML<br/>w_id Range: 2163 -</p> <p>w_id Min Warehouse:</p> <p>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 11390<br/>District id: 1<br/>Scale Down: No</p> | DRIVER07            | <p>Driver Engine:<br/>IIS Server: DL21<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa<br/>Protocol: HTML<br/>w_id Range: 6811 -</p> <p>w_id Min Warehouse:<br/>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 11850<br/>District id: 1<br/>Scale Down: No</p>  | DRIVER11            | <p>Driver Engine:<br/>IIS Server: DL23<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa<br/>Protocol: HTML<br/>w_id Range: 11551 -</p> <p>w_id Min Warehouse:<br/>w_id Max Warehouse:</p> <p>Scale: Normal<br/>User Count: 11850<br/>District id: 1<br/>Scale Down: No</p>   |
| tcp:15.1.104.1,1437 |  | tcp:15.1.104.1,1439 |  | tcp:15.1.104.2,1441 |  |
| 3301                |  | 7995                |  | 12735               |  |
| 1                   |  | 1                   |  | 1                   |  |
| 27402               |  | 27402               |  | 27402               |  |
| DRIVER04            | <p>Driver Engine:<br/>IIS Server: DL19<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa</p>  | DRIVER08            | <p>Driver Engine:<br/>IIS Server: DL21<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa</p>  | DRIVER12            | <p>Driver Engine:<br/>IIS Server: DL23<br/>SQL Server:</p> <p>Database: tpcc<br/>User: sa</p>  |
| tcp:15.1.104.1,1437 |  | tcp:15.1.104.1,1439 |  | tcp:15.1.104.2,1441 |  |

|                     |  |                     |  |                     |  |
|---------------------|--|---------------------|--|---------------------|--|
| 13920               | Database: tpcc<br>User: sa<br>Protocol: HTML<br>w_id Range: 12736 -    | tcp:15.1.104.2,1443 | SQL Server:<br>Database: tpcc<br>User: sa<br>Protocol: HTML<br>w_id Range: 17280 - | tcp:15.1.104.1,1443 | SQL Server:<br>Database: tpcc<br>User: sa<br>Protocol: HTML<br>w_id Range: 21674 - |
| 1                   | w_id Min Warehouse:  | 18268               | w_id Min Warehouse:  | 1                   | w_id Min Warehouse:  |
| 27402               | w_id Max Warehouse:  | 1                   | w_id Max Warehouse:  | 1                   | w_id Max Warehouse:  |
|                     | Scale: Normal<br>User Count: 11850<br>District id: 1<br>Scale Down: No | 27402               | Scale: Normal<br>User Count: 9890<br>District id: 1<br>Scale Down: No              | 27402               | Scale: Normal<br>User Count: 11850<br>District id: 1<br>Scale Down: No             |
| DRIVER13            | Driver Engine:   |                     | Driver Engine:   |                     | Driver Engine:   |
|                     | IIS Server: DL24<br>SQL Server:  | DRIVER17            | IIS Server: DL18<br>SQL Server:  | DRIVER21            | IIS Server: DL22<br>SQL Server:  |
| tcp:15.1.104.2,1442 | Database: tpcc<br>User: sa<br>Protocol: HTML<br>w_id Range: 13921 -    | tcp:15.1.104.1,1440 | Database: tpcc<br>User: sa<br>Protocol: HTML<br>w_id Range: 18269 -                | tcp:15.1.104.2,1444 | Database: tpcc<br>User: sa<br>Protocol: HTML<br>w_id Range: 22859 -                |
| 15105               | w_id Min Warehouse:  | 19349               | w_id Min Warehouse:  | 24043               | w_id Min Warehouse:  |
| 1                   | w_id Max Warehouse:  | 1                   | w_id Max Warehouse:  | 1                   | w_id Max Warehouse:  |
| 27402               | Scale: Normal<br>User Count: 11850<br>District id: 1<br>Scale Down: No | 27402               | Scale: Normal<br>User Count: 10810<br>District id: 1<br>Scale Down: No             | 27402               | Scale: Normal<br>User Count: 11850<br>District id: 1<br>Scale Down: No             |
| DRIVER14            | Driver Engine:   |                     | Driver Engine:   |                     | Driver Engine:   |
|                     | IIS Server: DL24<br>SQL Server:  | DRIVER18            | IIS Server: DL19<br>SQL Server:  | DRIVER22            | IIS Server: DL23<br>SQL Server:  |
| tcp:15.1.104.2,1442 | Database: tpcc<br>User: sa<br>Protocol: HTML<br>w_id Range: 15106 -    | tcp:15.1.104.1,1441 | Database: tpcc<br>User: sa<br>Protocol: HTML<br>w_id Range: 19350 -                | tcp:15.1.104.2,1445 | Database: tpcc<br>User: sa<br>Protocol: HTML<br>w_id Range: 24044 -                |
| 16290               | w_id Min Warehouse:  | 20488               | w_id Min Warehouse:  | 25228               | w_id Min Warehouse:  |
| 1                   | w_id Max Warehouse:  | 1                   | w_id Max Warehouse:  | 1                   | w_id Max Warehouse:  |
| 27402               | Scale: Normal<br>User Count: 11850<br>District id: 1<br>Scale Down: No | 27402               | Scale: Normal<br>User Count: 11390<br>District id: 1<br>Scale Down: No             | 27402               | Scale: Normal<br>User Count: 11850<br>District id: 1<br>Scale Down: No             |
| DRIVER15            | Driver Engine:   |                     | Driver Engine:   |                     | Driver Engine:   |
|                     | IIS Server: DL25<br>SQL Server:  | DRIVER19            | IIS Server: DL20<br>SQL Server:  | DRIVER23            | IIS Server: DL24<br>SQL Server:  |
| tcp:15.1.104.2,1443 | Database: tpcc<br>User: sa<br>Protocol: HTML<br>w_id Range: 16291 -    | tcp:15.1.104.1,1442 | Database: tpcc<br>User: sa<br>Protocol: HTML<br>w_id Range: 20489 -                | tcp:15.1.104.2,1446 | Database: tpcc<br>User: sa<br>Protocol: HTML<br>w_id Range: 25229 -                |
| 17279               | w_id Min Warehouse:  | 21673               | w_id Min Warehouse:  | 26413               | w_id Min Warehouse:  |
| 1                   | w_id Max Warehouse:  | 1                   | w_id Max Warehouse:  | 1                   | w_id Max Warehouse:  |
| 27402               | Scale: Normal<br>User Count: 9890<br>District id: 1<br>Scale Down: No  | 27402               | Scale: Normal<br>User Count: 11850<br>District id: 1<br>Scale Down: No             | 27402               | Scale: Normal<br>User Count: 11850<br>District id: 1<br>Scale Down: No             |
| DRIVER16            | Driver Engine:   |                     | Driver Engine:   |                     | Driver Engine:   |
|                     | IIS Server: DL25   | DRIVER20            | IIS Server: DL21   |                     |  |

DRIVER24  
 tcp:15.1.104.2,1447  
 27402  
 1  
 27402

Driver Engine:  
 IIS Server: DL25  
 SQL Server:  
 Database: tpcc  
 User: sa  
 Protocol: HTML  
 w\_id Range: 26414 -  
 w\_id Min Warehouse:  
 w\_id Max Warehouse:  
 Scale: Normal  
 User Count: 9890  
 District id: 1  
 Scale Down: No

10.05  
 5.00  
 10.05  
 5.00  
 10.05  
 5.00  
 10.05  
 5.00

Payment 10.00  
 2.01 0.10  
 Delivery 1.00  
 2.01 0.10  
 0.10  
 Stock Level 1.00  
 2.01 0.10  
 0.10  
 Order Status 1.00  
 2.01 0.10  
 0.10

Number of Parameter Sets: 3

AuditShutdown

| Think | Key  | RT     | RT    | Menu   | Txn          | Weight |
|-------|------|--------|-------|--------|--------------|--------|
| Time  | Time | Delay  | Fence | Delay  |              |        |
| 44.72 |      | 200.00 |       | 200.00 | New Order    | 0.10   |
|       |      | 5.00   |       | 0.10   |              |        |
|       |      | 200.00 |       | 200.00 | Payment      | 43.07  |
|       |      | 5.00   |       | 0.10   |              | 0.10   |
|       |      | 200.00 |       | 200.00 | Delivery     | 4.07   |
|       |      | 5.00   |       | 0.10   |              | 0.10   |
|       |      | 200.00 |       | 200.00 | Stock Level  | 4.07   |
|       |      | 20.00  |       | 0.10   |              | 0.10   |
|       |      | 200.00 |       | 200.00 | Order Status | 4.07   |
|       |      | 5.00   |       | 0.10   |              | 0.10   |

Audit

| Think | Key  | RT    | RT    | Menu  | Txn          | Weight |
|-------|------|-------|-------|-------|--------------|--------|
| Time  | Time | Delay | Fence | Delay |              |        |
| 44.96 |      | 12.05 |       | 18.00 | New Order    | 0.10   |
|       |      | 5.00  |       | 0.10  |              |        |
|       |      | 12.05 |       | 3.00  | Payment      | 43.01  |
|       |      | 5.00  |       | 0.10  |              | 0.10   |
|       |      | 5.05  |       | 3.00  | Delivery     | 4.01   |
|       |      | 5.00  |       | 0.10  |              | 0.10   |
|       |      | 5.05  |       | 3.00  | Stock Level  | 4.01   |
|       |      | 20.00 |       | 0.10  |              | 0.10   |
|       |      | 10.05 |       | 3.00  | Order Status | 4.01   |
|       |      | 5.00  |       | 0.10  |              | 0.10   |

~Default

| Think | Key  | RT    | RT    | Menu  | Txn       | Weight |
|-------|------|-------|-------|-------|-----------|--------|
| Time  | Time | Delay | Fence | Delay |           |        |
| 10.00 |      | 10.05 |       | 2.01  | New Order | 0.10   |
|       |      | 5.00  |       | 0.10  |           |        |



## Appendix D 60 Day Space Requirements

| TPC-C 60 Day Space Requirements |            |  |            |                |                  |                   |
|---------------------------------|------------|--|------------|----------------|------------------|-------------------|
| Warehouses                      | 30,000     |  |            |                | TpmC             | <b>344,928.15</b> |
| Table                           | Rows       | Data KB  | Index KB   | Extra 5% KB    | 8hr Space        | Total Space KB    |
| Warehouse                       | 30000      | 3200   | 152        | 168            |                  | 3520              |
| District                        | 300000     | 33336  | 216        | 1,678          |                  | 35230             |
| Customer                        | 900000000  | 654545456  | 40839128   | 34,769,229     |                  | 730153813         |
| History                         | 900000000  | 52554752   | 196400     |                | 9,400,706        | 52751152          |
| NewOrder                        | 270000000  | 4810696  | 12192      | 241,144        |                  | 5064032           |
| Orders                          | 900000000  | 29387760   | 14335248   |                | 12,296,009       | 43723008          |
| OrderLine                       | 8999970887 | 590162032  | 1390064    |                | 194,053,533      | 591552096         |
| Item                            | 100000     | 9416   | 168        | 479            |                  | 10063             |
| Stock                           | 3000000000 | 960000000  | 2024384    | 48,101,219     |                  | 1010125603        |
| Total                           |            | 2,291,506,648                                    | 58,797,952 | 83,113,917     | 215,750,249      | 2,433,418,517     |
|                                 | MB         |  |            |                |                  |                   |
| Dynamic Space                   | 656,352    | Sum of Data for Order, Orderline and History     |            |                |                  |                   |
| Static Space                    | 1,720,033  | Sum of Data+Index+5%-Dynamic Space               |            |                |                  |                   |
| Free Space                      | na         | Total Allocated Spac - ( Dynamic + Static Space) |            |                |                  |                   |
| Daily Growth                    | 120,744    | (Dynamic Space/(W*62.5))*tpmc                    |            |                |                  |                   |
| Daily Spread                    | -          | (Free Space - 1.5*Daily Growth) Zero Assumed     |            |                |                  |                   |
| 60 Day Space MB                 | 8,964,651  |  |            |                |                  |                   |
| 60 Day Space GB                 | 8,754.54   | GB   |            |                |                  |                   |
| Log Size                        | 400,000.00 | MB   |            |                |                  |                   |
| KB Per New Order                | 6.51       | KB   |            |                |                  |                   |
| 8 hr log MB                     | 1,052,045  | MB   |            |                |                  |                   |
| 8 hr log GB                     | 1,027.3876 | GB   |            |                |                  |                   |
|                                 |            | Disks  | Disks      | Formatted Size | Space            |                   |
| Space Usage                     | GB Needed  | Measured   | Size       | Size           | Available        |                   |
| 180 Day Space DB                | 8,754.54   | 756  | 36GB       | 33.918         | 25642.31         |                   |
| Total DB                        |            | 756  |            |                | <b>25,642.31</b> |                   |
| 8-hr log + mirror               | 2,054.78   | 36   | ??         | 279.397        | <b>10,058.29</b> |                   |
| OS, Swap                        | 3          | 2  | 36GB       | 33.918         | 67.84            |                   |
| Total Storage                   | 10,812.32  | GB   |            |                |                  |                   |
|                                 |            |  |            |                | 35,768.44        |                   |

## Appendix E 3<sup>rd</sup> Party Pricing

Microsoft Corporation Tel 425 882 8080  
 One Microsoft Way Fax 425 936 7329  
 Redmond, WA 98052-6399 <http://www.microsoft.com/>

**Microsoft**

July 10, 2006

Hewlett-Packard Company  
 Eric Deehr  
 One Microsoft Way  
 Redmond, WA 98052

Mr. Deehr:

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    |
|-------------|--|------------|----------|----------|
| 810-03134   | <b>SQL Server 2005 Enterprise Itanium Edition</b><br><i>Per Processor License</i><br><i>Discount Schedule: Open Program - No Level</i><br><i>Unit Price reflects a 6% discount from the retail unit price of \$24,999.</i> | \$23,432   | 4        | \$93,728 |
| P73-00285   | <b>Windows Server 2003 Standard Edition</b><br><i>Server License Only - No CALs</i><br><i>Discount Schedule: No Level</i><br><i>Unit Price reflects a 28% discount from the retail unit price of \$999.</i>                | \$719      | 8        | \$5,752  |
| 254-00170   | <b>Visual C++ Standard Edition</b><br><i>No Discounts Applied</i>  | \$109      | 1        | \$109    |
| N/A         | <b>Microsoft Problem Resolution Services</b><br><i>Professional Support</i><br><i>(1 Incident)</i>   | \$245      | 1        | \$245    |

All products are currently orderable through Microsoft's normal distribution channels. A list of these distribution channels can be found at <http://www.microsoft.com/products/info/render.aspx?type=mnpl&content=22%2flicensing&View=22>.

Defect support is included in the purchase price. Additional support is available from Microsoft PSS on an incident by incident basis at \$245 per call.

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](mailto:jamiere@microsoft.com).

Reference ID: PCjago0618058247.

Please include this Reference ID in any correspondence regarding this price quote.



**AdderView Matrix MP 16-Port kvm switch**  
16 computers, 2 users .. rackmountable & cascable

**Free Shipping Everyday**

**We'll Beat Any Price**

|                              |                                |                               |                             |                                |                                 |
|------------------------------|--------------------------------|-------------------------------|-----------------------------|--------------------------------|---------------------------------|
| <a href="#">Brands</a>       | <a href="#">KVM Switch</a>     | <a href="#">Extenders</a>     | <a href="#">KVM over IP</a> | <a href="#">KVM over CAT5</a>  | <a href="#">KVM Cables</a>      |
| <a href="#">Video Switch</a> | <a href="#">Video Splitter</a> | <a href="#">Remote Reboot</a> | <a href="#">KVM Drawer</a>  | <a href="#">Console Switch</a> | <a href="#">Computer Cables</a> |

[Order Tracking](#) : [Shopping Cart](#) : [Customer Care Center](#) : [Technical Help Center](#)

## Shopping Cart

| Delete           | Accessories            | No.  | Product   | Quantity | Price     | Extended      |
|------------------|------------------------|------|---|----------|-----------|---------------|
|                  | <a href="#">Add-On</a> | 5203 | <a href="#">Aten MasterView Plus 8 Port KVM Switch w/Cables</a> | 1        | \$ 199.00 | 199           |
| <b>Sub Total</b> |                        |      |   |          |           | <b>199.00</b> |

### Contact

Toll-Free:  
**1-800-KVM-Switch**  
( 1-800-586-7948)  
Int'l/local:  
1.770.971.1924  
Fax:  
1.770.971.2639  
Email:  
[sales@kvms.com](mailto:sales@kvms.com)  
Mail:  
PO Box 669875  
Marietta, GA 30066-0115

**Check-Out**



|  |  |
|--|--|
| <h3>XpressTrack</h3> <p>Enter XpressTrack Code (if any)</p> <input type="text"/> <input type="button" value="GO"/> | <h3>PromoCode</h3> <p>Enter promotional Code (if any)</p> <input type="text"/> <input type="button" value="GO"/> |
|--|--|

### Instructions

- To increase the quantity, just select the quantity in the quantity box.  
Quantity
- If you like to remove an item, click on the Trash Can next to the item
- Please review your

shopping cart.

- Once ready, click Check-Out to place your order.
- If you like to order by phone, Call us Toll-Free 1.888.586.7948 and reference this code **POR-434422**

#### Payments Accepted

- **Credit Cards**  
Visa, Master Card, Amex and Discover.
- **Check On-Line**  
Pay directly from checking account. (US Customers only)
- **COD (Cash on Delivery)**  
Pay when you receive your shipment (US Customers only)
- **Purchase Order**  
US Business and government can pay by purchase order.
- **Wire Transfer**  
International customers pay by wire transfer.

[Privacy Policy](#) | [Security](#) | [Returns Policy](#) | [Shipping Information](#) | [About us](#)

1996 - 2005 © KVMs.com, all rights reserved. Brand names are trademarks of their respective holders

# DVI KVM Switches



[ORDER TRACKING](#)



[CART](#)

[CONTACT US](#)

[ORDER CHECKOUT](#)

Your cart contains:

| Quantity | Item #                       | Description                             | Options | Price     | Actions                |
|----------|------------------------------|---|---------|-----------|------------------------|
| 6        | <a href="#">PRIME-SB5200</a> | PRIME SVCAGR 1YR 7X24X4 RESP FOR SB5200 |         | \$6821.40 | <a href="#">Remove</a> |

Recalculate

**Total:** **\$6821.40** *Tax not*

Sales tax will be added for orders shipped to NY and MA. **OVERNIGHT ORDERS:** Submissions after 3pm will be shipped **NEXT**

\*\* *Exact amount to be calculated after shipping is*

[Keep](#)

[Checkout...](#)

[Home](#)

[Cart](#) | [Checkout](#) | [Customer Registration](#) | [Menu](#)



**866.463.3372**  
M-F 8am-6pm [EST]

SEARCH

PRODUCTS

*HBA's*

- QLogic >
- QLogic - EMC >
- Emulex >
- Emulex - EMC >
- Adaptec >

*Switches*

- QLogic >

*Transceivers*

- Finisar
- Cisco
- Agilent

*Cables*

- LC-LC
- SC-LC
- SC-SC

Credit Application

CHECKOUT

2 x QLogic SANbox  
5200 16-Port  
Enabled Full Fabric  
Switch, (16) SFPs

\$10,384.00

--- View Cart ---



Secured By:

## Shopping Cart Contents

| Remove                   | Product(s)   | Qty.                           | Total       |
|--------------------------|--|--------------------------------|-------------|
| <input type="checkbox"/> |  QLogic SANbox 5200 16-Port Enabled Full Fabric Switch, (16) SFPs | <input type="text" value="2"/> | \$10,384.00 |

Sub-Total:\$10,384.00

UPDATE CART

CONTINUE SHOPPING

CHECKOUT





**866.463.3372**  
M-F 8am-6pm [EST]

SEARCH

PRODUCTS

**HBAs**

- QLogic >
- QLogic - EMC >
- Emulex >
- Emulex - EMC >
- Adaptec >

**Switches**

- QLogic >

**Transceivers**

- Finisar
- Cisco
- Agilent

**Cables**

- LC-LC
- SC-LC
- SC-SC

Credit Application

CHECKOUT

1 x QLogic SANblade  
QLA2342-CK  
\$1,825.00

--- View Cart ---



Secured By:



## Shopping Cart Contents

| Remove                   | Product(s)   | Qty. | Total      |
|--------------------------|--|------|------------|
| <input type="checkbox"/> |  <b>QLogic SANblade QLA2342-CK</b><br>- Warranty Upgrade 5 Yr Standard Warranty | 1    | \$1,825.00 |

Sub-Total: \$1,825.00


UPDATE CART

CONTINUE SHOPPING

CHECKOUT





|  <b>Hewlett Packard Company</b>   |           | <b>HP Integrity rx6600</b> |            | <b>TPCC Rev 5.7 April 2006 Pricing 1.1.0<br/>July 18 2006</b> |                      |                  |
|--|-----------|----------------------------|------------|---|----------------------|------------------|
| Description  | Price Key | Part Numbr                 | Unit Price | Qty   | Extended Price       | 3 Yr Maint Price |
| <b>HP Integrity rx6600 with (4) 1.6GHz/24MB Processor</b><br>includes dual port 10/100/1000GbE adapter and 1 power supply  |           |                            |            |   |                      |                  |
| I/O backplane  | 1*        | AD134A#180                 | \$43,845   | 1   | \$43,845             |                  |
| core I/O: 8-port SAS Smart Array RAID Controller Card  | 1*        | AD296A                     | \$0        | 1   | \$0                  |                  |
| 192GB - 16GB DDR2 memory quad (4x4GB)  | 1*        | AB036A #100                | \$500      | 1   | \$500                |                  |
| 48 DIMM Carrier Board  | 1*        | AB566A                     | \$18,977   | 12  | \$227,724            |                  |
| 36GB, 10K rpm SAS HDD  | 1*        | AD127A                     | \$4,495    | 1   | \$4,495              |                  |
| Racked form factor kit   | 1*        | AD140A                     | \$382      | 2   | \$764                |                  |
| 3 Year Support (Hardware and Software)   | 1*        | AD053A                     | \$150      | 1   | \$150                |                  |
| DVD-ROM  | 1*        | HA110A3                    | \$10,947   | 1   |                      | \$10,947         |
| FC-HBA 2GB, 2 Channel (LP1050)   | 1*        | AD142A                     | \$230      | 1   | \$230                |                  |
| 36GB, 15krpm Ultra320 disk   | 1         | AD168A                     | \$2,000    | 1   | \$2,000              |                  |
| 36GB, 15krpm Ultra320 disk (10% Spares)  | 1         | 286776-B22                 | \$269      | 756   | \$203,364            |                  |
| Storageworks MSA30 SB  | 1         | 286776-B22                 | \$269      | 76  | \$20,444             |                  |
| Storageworks MSA30 SB (10% Spares)   | 1         | 302969-B21                 | \$2,829    | 36  | \$101,844            |                  |
| HP Rack 5642   | 1         | 302969-B21                 | \$2,829    | 4   | \$11,316             |                  |
| UPS - HP R1500 XR Low Voltage US   | 1         | 358254-B21                 | \$689      | 5   | \$3,445              |                  |
| HP 16A High Voltage Modular PDU  | 1         | 204404-001                 | \$866      | 1   | \$866                |                  |
| Modular Storage Array 1000   | 1         | 252663-B24                 | \$299      | 10  | \$2,990              |                  |
| Modular Storage Array 1000 (10% spares)  | 1         | 201723-B22                 | \$6,995    | 19  | \$132,905            |                  |
| MSA1000 controller (10% spares)  | 1         | 201723-B22                 | \$6,995    | 2   | \$13,990             |                  |
| HP 300GB 10 KRPM U320  | 1         | 218231-B22                 | \$4,290    | 2   | \$8,580              |                  |
| HP 300GB 10 KRPM U320(10% Spares)  | 1         | 350964-B22                 | \$699      | 8   | \$5,592              |                  |
| 5M LC to LC Cable Kit  | 1         | 350964-B22                 | \$699      | 2   | \$1,398              |                  |
| TA5300 Enclosure for DAT tape  | 1         | 221692-B22                 | \$77       | 19  | \$1,463              |                  |
| DAT Tape   | 1         | C7508B                     | \$729      | 1   | \$729                |                  |
|  | 1         | C7497B                     | \$1,049    | 1   | \$1,049              |                  |
| <b>Server Subtotal</b>   |           |                            |            |   | <b>\$789,683</b>     | <b>\$10,947</b>  |
| <b>Microsoft Windows Server 2003, Enterprise Edition for Itanium-based Systems, SP1</b>  |           |                            |            |   |                      |                  |
|  | 1         | T2030A                     | \$2,149    | 1   | \$2,149              |                  |
| <b>Server Software Subtotal</b>  |           |                            |            |   | <b>\$2,149</b>       | <b>\$0</b>       |
| <b>DL140 G2 3.6GHz Xeon 1GB 80GB SATA</b>  |           |                            |            |   |                      |                  |
| 2nd 3.6GHz Xeon Processor for DL140  | 1         | 383504-001                 | \$2,149    | 8   | \$17,192             |                  |
| 3 Year Support (ProLiant Hardware)   | 1         | 378283-B21                 | \$999      | 8   | \$7,992              |                  |
| HP Mouse   | 1         | HA110A3                    | \$419      | 8   |                      | \$3,352          |
| HP Enhanced Keyboard (USB/PS2)   | 1         | P5304M                     | \$28       | 8   | \$224                |                  |
| HP ProCurve 2824 port switch   | 1         | DC852A#ABA                 | \$25       | 1   | \$25                 |                  |
| 3 Year Support (ProCurve Hardware)   | 1         | J4903A                     | \$2,499    | 1   | \$2,499              |                  |
| S7540 17in CRT Monitor   | 1         | HA110A3                    | \$1,041    | 1   |                      | \$1,041          |
|  | 1         | PF997AA                    | \$139      | 1   | \$139                |                  |
| <b>Client Subtotal</b>   |           |                            |            |   | <b>\$28,071</b>      | <b>\$4,393</b>   |
| <b>**Total Extended Price:</b>   |           |                            |            |   | <b>\$835,243</b>     |                  |
| <b>**Total Discount:</b>   |           |                            |            |   | <b>-\$193,544</b>    |                  |
| <b>HP's Large Configuration Discount **</b>  |           |                            |            |   |                      |                  |
| <b>Price Key: 1-HP</b>   |           |                            |            |   |                      |                  |
| * Not immediately orderable. See Appendix E for details  |           |                            |            |   |                      |                  |
| ** A 35.41% discount was based on the overall value of the specific components from HP (Price Key 1) in this single quotation. Discounts for similarly sized configurations will be similar to those quoted here, but may vary based on the components in the configuration  |           |                            |            |   |                      |                  |
| <b>3 year cost of ownership:</b>   |           |                            |            |   | <b>\$641,699 USD</b> |                  |
| 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 section of the TPC benchmark specification. If you find that the state prices are not available to these terms, please inform the TPC at pricing@tpc.org. Thank you |           |                            |            |   |                      |                  |
| Sales contact Vendor 1: HP Sales Development, 19111 Pruneridge Ave., Cupertino, CA 95014 (408) 447 2320 Sales contact<br>Vendor 2: Jamie Reding (425) 703-0510 jamiere@microsoft.com<br>Vendor 3 - HBACentral.com Vendor 4 - KVMs.com Vendor 5 - softwareforless.com   |           |                            |            |   |                      |                  |

The following product(s) will be orderable as of 9/1/2006:

- HP Integrity rx6600 with (4) 1.6GHz/24MB Processor includes dual port 10/100/1000GbE adapter and 1 power supply - AD134A#180
- I/O backplane - AD296A
- core I/O: 8-port SAS Smart Array RAID Controller Card - AB036A #100
- 48 DIMM Carrier Board - AD127A
- 36GB, 10K rpm SAS HDD - AD140A
- Racked form factor kit - AD053A
- 3 Year Support (Hardware and Software) - HA110A3
- DVD-ROM - AD142A

The following product(s) will be orderable as of 12/1/2006:

- 192GB - 16GB DDR2 memory quad (4x4GB) - AB566A

For Pricing Verification and Ordering, Contact:

HP Sales Development, 19111 Pruneridge Ave., Cupertino, CA 95014 (408) 447 2320